NOVEMBER 2019

intercom

THE MAGAZINE OF THE SOCIETY FOR TECHNICAL COMMUNICATION

CONTENT ENGINEER ROLES AND RESPONSIBILITIES 8

FOLLOW YOUR NORTH STAR TO A SUCCESSFUL CONTENT ENGINEERING TEAM 12

ENGINEERING CONTENT FOR SUPERIOR SEARCH PERFORMANCE: INTRODUCING STRUCTURED DATA 18

WANT PERSONALIZATION? START WITH CONTENT ENGINEERING 22

THE ROLES OF STANDARDS AND GOVERNANCE IN ENGINEERING CONTENT FOR MULTICHANNEL DISTRIBUTION 26

AUTONOMOUS AGILE TEAMS AND ENTERPRISE CONTENT STRATEGY: AN IMPOSSIBLE COMBO? 31

PRODUCT ANSWERS: ENGINEERING CONTENT FOR FRESHNESS 35

CONTENT ENGINEERING

All-in-one

Deliver a consistent experience at every stage of the customer journey by bringing marketing and technical content onto a single platform, with **XML Documentation for Adobe Experience Manager**.

bit.ly/DITA-CCMS

Get in touch

techcomm@adobe.com

+1-408-536-1144

Adobe DITA World 2019

Register now for the world's biggest DITA online conference.

www.dita-world.com

© 2019 Adobe Inc. All rights reserved.

November 2019

Volume 66 | Issue 7

4 **From the Editor**

6 **From the Guest Editor**

FEATURES

8 **Content Engineer Roles and Responsibilities**
By Ann Rockley
Ann defines the role of the content engineer and its importance to complete modern content teams.

12 **Follow Your North Star to a Successful Content Engineering Team**
By Carolyn Swift-Muschott
Carolyn examines when the time is right to create a formal content engineering team for your organization.

18 **Engineering Content for Superior Search Performance: Introducing Structured Data**
By Aaron Bradley
Aaron explains why structured data is crucial for increasing the searchability and discoverability of content.

22 **Want Personalization? Start with Content Engineering**
By Lisa Trager
Personalization of content needs content engineering. Lisa shares her experiences managing complex, ever-growing content challenges.

26 **The Roles of Standards and Governance in Engineering Content for Multichannel Distribution: Why Standards Are More Beneficial than You Think**
By Ulrike Parson
Ulrike looks at how public standards, rather than being restrictive, can actually make content more suitable for omnichannel distribution.

31 **Autonomous Agile Teams and Enterprise Content Strategy: An Impossible Combo?**
By Tom Johnson
Tom explores the differences between independent and unified documentation teams and what solutions might have a positive impact on the content lifecycle.

35 **Product Answers: Engineering Content for Freshness**
By Megan Gilhooly
Megan examines the benefits of fresh content and the role structured content plays in its delivery.

SOCIETY PAGES

38 Registration Is Open for the 2020 STC Technical Communication Summit & Expo

38 Subscribe to STC Roundtable

39 Nominations Open for 2020 Community Achievement and Pacesetter Awards

39 Nominations Open for the 2020 Jay R. Gould Award

COLUMNS

CONTENT ENGINEERING

40 Defining Intelligent Customer Experience

By Cruce Saunders

ASK A TECH COMM MANAGER

44 Providing Value in an Agile Environment as a Technical Writer

By Cindy Currie and Kit Brown-Hoekstra

THE COGNITIVE COMMUNICATOR

45 Spaces, Doorways, and Usable Designs

By Kirk St.Amant

MEET THE CHANGE AGENTS

47 The Components of Content: Meet Patrick Bosek

By Scott Abel

DEPARTMENTS

FYI

51 Mark Your Calendar

Organization Events Across the Globe

OFF HOURS

52 Wearing Many Hats

By Surag Ramachandran

ADVERTISERS

C2	Adobe
28, C3	STC Certification
3, 50	STC Education
46	STC Membership
43	STC Summit & Expo 2020
C4	Whatfix

Experience Professional Growth with STC Education!

STC offers a wide variety of online education options throughout the year. Whether you need an introduction to a subject, an in-depth review, or just a brush-up, STC has what you need. Advance your career with STC's varied collection of online education.

Live Weekly Webinars

Multi-Week Online Courses

Recorded Webinars

Free On-Demand Seminars
(Members Only)

Visit stc.org/education

intercom

Intercom, the magazine of the Society for Technical Communication, is published to provide practical examples and applications of technical communication that will promote its readers' professional development.

Publisher Society for Technical Communication
Chief Executive Officer Elizabeth E. (Liz) Pohland
Executive Editor Andrea L. Ames
Assistant Editor James Cameron

CONTRIBUTING EDITORS
Scott Abel, *Meet the Change Agents*
Thomas Barker, *The Academic Conversation*
Kit Brown-Hoekstra and Cindy Currie, *Ask a Tech Comm Manager*
Cruce Saunders, *Content Engineering*
Michelle Corbin, *Editing Matters*
Jeanette Evans and Charles Dull, *Emerging Technologies*
Mark Lewis, *Metrics*
Alan Porter, *Convergence Conversations*
Kirk St.Amant, *The Cognitive Communicator, Health and Medical Communication*
Ryan Weber, *Student Perspectives*
Russell Willerton, *Ethics*

STC BOARD OF DIRECTORS
Officers
Ben Woelk, *President*
Craig Baehr, *Vice President*
Jane Wilson, *Immediate Past President*
James Bousquet, *Treasurer*
Kirsty Taylor, *Secretary*
Directors Bethany Aguad, Alisa Bonsignore, Todd DeLuca, Laura Palmer

EDITORIAL ADVISORY COMMITTEE
Kit Brown-Hoekstra, Chair
Saul Carliner
Michelle Corbin
Michael Opsteegh
Kirk St.Amant

SOCIETY FOR TECHNICAL COMMUNICATION
3251 Old Lee Highway, Suite 406
Fairfax, VA 22030
+1(703) 522-4114 | +1(703) 522-2075 fax
Send questions and correspondence to the editor at the above address. General information about STC: stc@stc.org *or* www.stc.org.

Address Changes: membership@stc.org
Intercom Editorial: intercom@stc.org

Reprint Permissions: www.copyright.com
Intercom (ISSN 0164-6206) is published 8 times a year (monthly, except for February, April, June, and August) by the Society for Technical Communication, 3251 Old Lee Highway, Suite 406, Fairfax, VA 22030, a nonprofit educational organization, as a service to its membership. Membership in STC includes an online subscription to **Intercom**. Periodicals postage paid at Fairfax, VA, and additional offices.
Postmaster Send address changes to **Intercom**, Attn: Membership, c/o Society for Technical Communication, 3251 Old Lee Highway, Suite 406, Fairfax, VA 22030.
Copyright © 2019 by the Society for Technical Communication. All rights reserved. Printed in the USA. All articles in **Intercom** are copyrighted by the authors unless otherwise indicated. Anyone interested in reprints, reproductions, or reuse must contact the authors for permission.
To advertise in **Intercom**, contact *intercom@stc.org*, Society for Technical Communication, 3251 Old Lee Highway, Suite 406, Fairfax, VA 22030. Ph: +1(703) 522-4114.

A Note from the Editor

I AM TECHNOLOGY-AGNOSTIC. As a content strategist, experience architect and designer, organization designer, change manager, audience researcher and builder, culture builder, and leadership coach, I don't care which authoring tool, content management system, markup language, etc., my clients use. And nothing saddens me more than to see content practitioners, managers, and executives throw tools, exclusively, at problems.

I do, however, care very much that the right infrastructure, tools, and technologies are employed to deliver the right experience that will enable the wild success of customers and the companies who serve those customers. And during the past couple of decades of my career, I've been lucky enough to work with amazing content technology practitioners—now called *content engineers*—who, depending on the day or the situation, were the yin to my yang, or the yang to my yin.

These content engineers by other names and titles care just as deeply that customers and companies are successful. They don't tend care as much as I do what specific strategies, designs, and experiences get them to that success, but they do care that the technology delivers those experiences and enables that success flawlessly.

I've always found my partnership, collaboration, and relationship with content engineering to be so enlightening, rewarding, synergistic, and multiplicative in terms of what we could achieve and deliver together that I thought it time to highlight this practice in *Intercom*. And I couldn't think of anyone better to guest edit the issue than my good friend Cruce Saunders. When I met him after hearing him speak at the Intelligent Content Conference several years ago, Cruce was the first person I ever heard put a name to the practice of content engineering.

I've always found my partnership, collaboration, and relationship with content engineering to be so enlightening, rewarding, synergistic, and multiplicative in terms of what we could achieve and deliver together that I thought it time to highlight this practice in *Intercom*.

Cruce is a speaker, author, and entrepreneur who—as you will see when you read the rest of this issue—is passionate about content engineering. He has a great understanding of, and respect for, all of the activities (and the people performing those activities) that are required to bring the right information from large, complex libraries of content to the right consumer in the right experience. He is a thought leader in this space, and his north star is friction-free customer success. He has his thumb on the pulse of the industry and his ear to the words of thought leaders, and many of those thought leaders also follow him. As a result, he is able to bring their experiences and

knowledge to us. Thank you to Cruce and to the feature authors of this issue: Ann Rockley, Carolyn Swift-Muschott, Aaron Bradley, Lisa Trager, Ulrike Parson, Tom Johnson, and Megan Gilhooly.

In this issue, you'll also find an excellent selection of columns from some of our regular columnists:

- Cruce caps off the content engineering discussions in his "Content Technology" column.
- Cindy Currie and Kit Brown-Hoekstra share their tips on providing value as a technical communicator in an agile environment in "Ask a Manager."
- Kirk St.Amant describes physical spaces and spatial cognition—and how that applies to the design of online interfaces in "The Cognitive Communicator."
- Scott Abel interviews Patrick Bosek for "Meet the Change Agents" and adds markup languages and component content management systems to the technology discussion in this issue.

As always, don't forget to check out the Society Pages to keep up with what's going on around STC!

Interested in learning more about content engineering? Want to respectfully argue with a point made by an author? Catch us online at *https://www.stc.org/intercom/*. Just click on an article's title to view it, scroll to the bottom of the page, and comment! We'd love to hear from you! You can also contact me, Cruce, or any of the authors or columnists using the contact info in their article bios. We look forward to having a conversation with you!

Until next time, to your technological success!

Andrea Ames

— Andrea L. Ames
andrea@idyllpoint.com

Content Engineering and the Next Generation of Customer Experience

EVERY DAY, WE SEE GLIMMERS of our organization's future customer experiences. Those insights come in the interactions we have as consumers ourselves. We see bits of the future when we:

- Make purchases on Amazon from personalized suggestions
- Interact with smart Alexa and Google services via voice
- Find our room preferences already handled at a hotel
- Receive flight alerts and gate changes through app push notifications
- Book and pay for rides or rooms in a couple of quick taps
- Receive emails with relevant product and media recommendations
- Digitally—and with ease—handle previously complex personal finances and investments
- Interact in rich and multisensory experiences in games and dialogues
- Find spot-on healthcare answers and recommendations across multiple interfaces
- Resolve basic service issues within a chatbot on a mobile app

Everywhere we look, it's getting easier to consume products and services. Sure, there are plenty of failures. But overall, the trend in consumer technology is toward streamlined, personalized ease.

How about customer experiences across the business-to-business (B2B) enterprise? Not so smart. It's often the same old nightmare to use documentation, dig out answers to technical support questions, find and use relevant product specifications, or transfer information from supplier to internal format. Even when consuming marketing material, while we find enterprise websites faster-loading and more visually impressive than ever and with more content than ever, those sites are still painful to use. Bad experiences are not habit-forming. Inflicting pain on customers as they attempt to find information, process transactions, and resolve problems inevitably leads to loss of market share versus competitors that customers find easier to work with.

Compared to smooth consumer-like experiences, enterprise digital interactions now feel like a productivity drain. Old-school websites now feel heavy. This heaviness is not isolated to only the B2B enterprise, although the problems are most pronounced there. Many large, consumer-facing giants suffer from enterprise content malaise, seemingly not even attempting to streamline and personalize the customer experience. They remain stuck in archaic content management regimes and processes. According to a 2018 Forrester report, 80 percent of companies believe content supply chain challenges impede their ability to deliver on business objectives. The problem is a content problem, and the answer, as we will explore, includes content engineering.

It Takes a Village: Strategy, Engineering, Operations

Multiple roles are involved in delivering intelligent customer experiences. Content strategy, content engineering, and content operations are three focal practices that enable, facilitate, and orchestrate the supply chain that produces these next-generation interactions. The emergence of established content strategy functions and the widespread adoption of related roles across enterprises has helped to solve some of these challenges. Content operations and content engineering are as foundational and distinct as content strategy, and all practices must be represented in organizations striving to meet customers with contextually relevant content.

For most organizations, there is already some form of content strategy and operations happening—whether those titles are used or not. Although each role is vital, in this issue we will mainly focus on the content engineering practice and learning from practitioners that have been incorporating content engineering principles into enterprise content environments.

Omnichannel content leaders, regardless of department, need to get intentional about incorporating engineering practices into the overall content operating model. Without content engineering disciplines incorporated into content teams, it's possible to have big, painful gaps between strategy and execution.

On the larger wish list is building content strategy, engineering, and operation practices into a chartered orchestration function, giving cross-silo teams a way to share and integrate content. Even without a large portfolio of responsibility, however, the existence of these practices within any organization will advance content. Together these practices can form a content services organization (CSO). [A] recently published a white paper reviewing the CSO structure and functions in detail, but suffice to say, the practice of content engineering plays a pivotal role.

A Personal History with Content Engineering

Content has always been in my blood. I gave my first STC talk, introducing the internet as a publishing medium, in the late 1990s for the STC Dallas Chapter. Fast-forward to years of running a company that evolved from Web firm, to interactive agency, to CMS platform startup, to integrator of third-party WCMS products, until founding [A] to focus entirely on intelligent content in 2012.

Inspired by Joe Gollner's blog, *The Content Philosopher;* Ann Rockley's work; and technical communication professionals working on various approaches to content portability, my mind lit up realizing all the connections to content types *outside* of documentation and support. Content engineering was born in the interplay and expressed so well in Joe's blog and Ann's work.

I compared the content modeling and metadata specifications work we accomplished for marketing content sets focused on dynamic Web publishing and search engine discoverability and realized just how close (and yet how far away) were the basic patterns to the single-source publishing approaches in technical communication. A bridge was missing and needed.

In reading various Web content strategists' definitions, it was clear that we needed a distinct practice, with disciplines including model, metadata, markup, schema, taxonomy, and organization topology. We needed a practice focused not on the "content of the content" but on the structure and semantics of the content. I gave my first public talk on the role of content engineering at the 2015 Intelligent Content Conference in San Francisco.

Since then, [A] has worked with and interviewed dozens of senior executives, more than one hundred directors and senior managers, and hundreds of authors across enterprise content teams. Everyone agrees we need a new way to "do" content, and more and more of us agree that content engineering must be a clear part of the solution.

Converged Content, Converged Supply Chains

Content shared between marketing, sales, technical communication, knowledge management, learning, localization, and others contributes to the unification of pre- and post-sale customer experiences: Customers now *choose* where to start and where to go next within the content sets.

Given the complexity of enterprise content publishing, the practice of content strategy alone cannot solve today's challenges. Content strategists need specialized counterparts and enablers to realize and sustain the goals of a more strategic deployment of content assets. This includes the integration of engineering and operations functions.

Engineered content assets can be used in many places at once: they may be related, discovered, and used to deliver value across multiple systems and platforms, when and where needed, at the fastest possible throughput.

Content engineers ensure that content structures, semantics, and technologies are in place to take the content strategy from plan to a technically realized reality. Content engineers become a resource for talented and often isolated members of technical communication or content marketing teams, bridging worlds and content sets through tactical tools, like content and semantic models, that encourage the collaboration needed to deliver integrated, intelligent experiences.

Content Engineering in This Issue

As the guest editor for this content engineering-themed issue of *Intercom*, it has been my distinct pleasure to work with authors selected from among the top professionals leading

systematic change to content. All of them have, in ways large and small, incorporated the various disciplines within the practice of content engineering into their thinking, work, and planning.

It's a power-packed issue! Many thanks to the amazing leaders—bringing decades of experience to the conversation—featured in this issue for sharing their time and insights with *Intercom* readers!

Ann Rockley's article, "Content Engineer Roles and Responsibilities," focuses on defining the role of a content engineer and why they are necessary to complement and complete modern content teams. This role is juxtaposed with the content strategist, a more tenured role in the enterprise, and addresses the background, responsibilities, and path to become a content engineer.

Carolyn Swift-Muschott of Cengage explores when the time is ripe for a formal content engineering team in "Follow Your North Star to a Successful Content Engineering Team." Outlining the all-too-familiar pain points many face in the industry today, Carolyn illustrates the skills content engineers use to orchestrate successful, dependable content delivery. Learn from one of the early leaders who has recruited and integrated a team of content engineers about the profile, mindset, and team composition that produce results.

Smart search and instantaneous discoverability happens when we structure content to support discovery. In "Engineering Content for Superior Search Performance: Introducing Structured Data," Electronic Arts' Aaron Bradley takes readers through why structured data is crucial for relevant usable data, the standards in place today that support machine readability, and how to increase search and discoverability of content. Learn where engineering content for search performance starts.

In "Want Personalization? Start with Content Engineering," the title speaks for itself. Few that confront personalization projects would argue with the need for content engineering. Hear long-time practitioner Lisa Trager share experiences from the deep within silos, and some of the fundamental patterns needed for managing complex, ever-growing content challenges.

Assembling structured content modules showing relevant content, in context, at precisely the right moment, calls for engineering. Ulrike Parson examines "The Roles of Standards and Governance in Engineering Content for Multichannel Distribution" and gives readers navigational assistance in the form of public standards along the path toward intelligent customer experiences. Understand how incorporating standards helps make content suitable for omnichannel distribution.

But how do we balance agile development approaches and enterprise-wide structured content engineering? Tom Johnson of Amazon explores this difficult question in "Autonomous Agile Teams and Enterprise Content Strategy; An Impossible Combo?" There are benefits to both independent and unified documentation practices in authoring and managing content, and the harmony of them can seem impossible. Tom weighs both sides and proposes a sensible, and most importantly, practical solution that positively impacts the full content lifecycle.

Once content is structured, new possibilities emerge to reduce the cost and time of creation while maintaining fresh content for customers. In "Product Answers: Engineering Content for Freshness," Megan Gilhooly explores the benefits of fresh content and the role structured content plays in delivery. See how a designed content architecture, with modular topics and leveraging content reuse creates lasting, impactful, and relevant content.

—Cruce Saunders
cruce@simplea.com

REGISTER FOR UPCOMING ONLINE COURSE (6 CEUs)

Introductory Markup and Scripting with HTML and CSS

9 January–13 February 2020

Thursdays | 2:00 – 3:30 PM EST

INSTRUCTOR:
Dr. Craig Baehr

OVERVIEW:
This six-session course will introduce participants to introductory markup and scripting with HTML and CSS. The course will also introduce students to readings, resources, and tools available to help develop basic coding skills through practice.

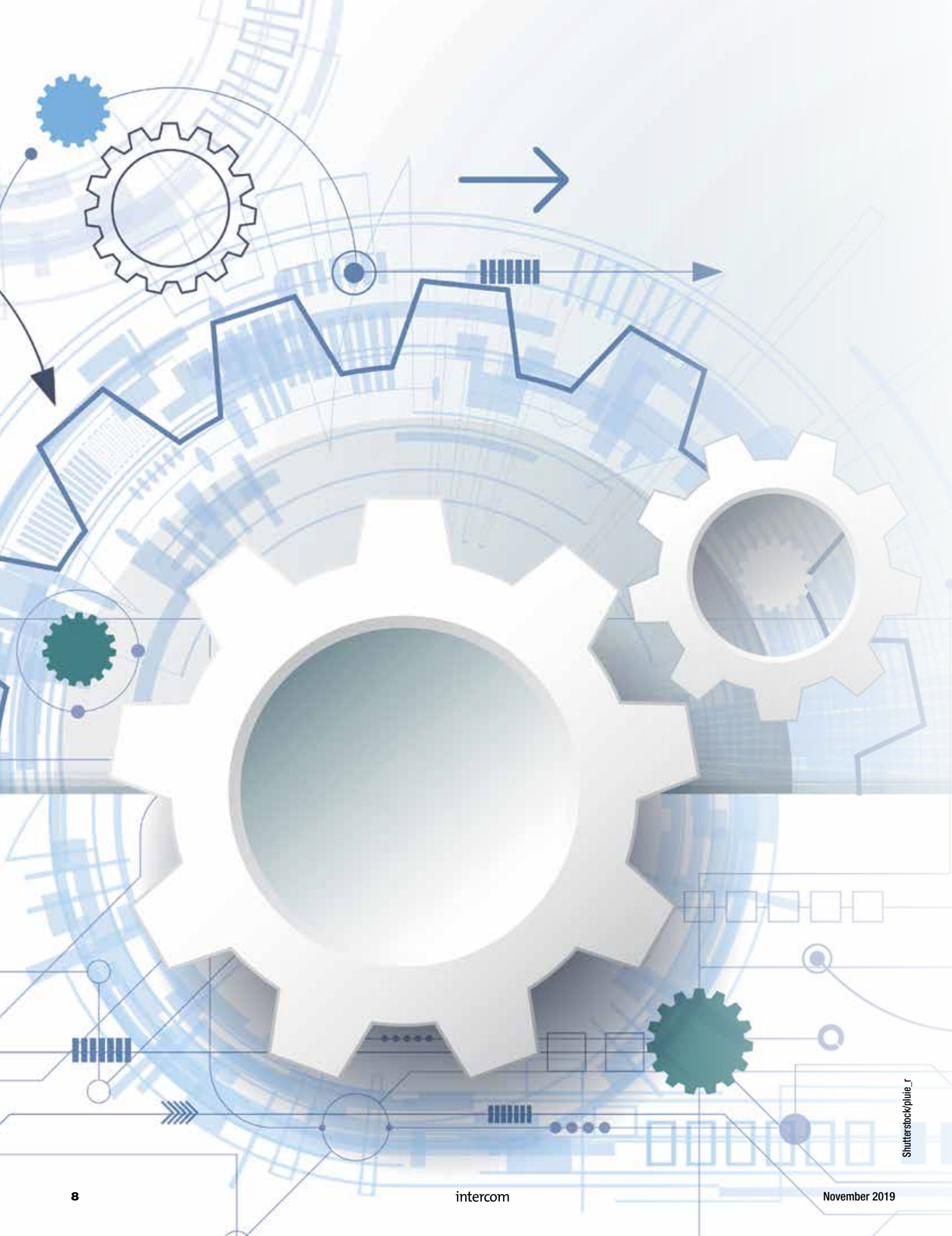
Each session will feature a different topic (HTML Syntax, Working with Text Editors, Tables and Semantic Markup, Form Building, Integrating HTML and CSS, CSS Syntax, Working with Positional CSS) and will also challenge students with weekly practice assignments to help them develop their coding skills. Students also will be encouraged to present their work in progress and receive constructive feedback from the instructor and their peers.

The goal of this course is to provide students with an understanding of the wide range of skills, methods, and techniques used basic content markup and scripting for the Web. The course is also designed to prepare students for more advanced or specialized study in other topics related to markup and scripting in Web development.

COST:
Members: $595
Nonmembers: $995
Student Members: $295

STC

www.stc.org



Shutterstock/pluie_r

Content Engineer Roles and Responsibilities

By ANN ROCKLEY | *STC Fellow*

CONTENT STRATEGY and content engineering are two practices that have been growing and evolving along with enterprise content. This article focuses on the role and responsibilities of the content engineer.

Why Do We Need Content Engineers?

To deliver content that meets customers' personalized needs, in multiple channels, on multiple devices, and in multiple languages is not a simple task. It takes someone who has an in-depth understanding of content and technical expertise to make it happen. A content engineer works in concert with a content strategist to support content in the multiverse of audiences, channels, personalization, localization, and technologies. The two roles are intertwined but focus on slightly different facets of the content experience. To differentiate, let's start with the role of the content strategist.

Content strategists typically focus on the customer experience, the content itself, and editorial guidelines. A content strategist is responsible for designing or deciding:

- Customer personas
- Customer journeys
- What topics to address
- When to support the customer at multiple points in the customer journey
- The best content types (text, visuals, video)
- SEO guidelines to ensure that people searching online can find the content
- Style guidelines on how to write for the audience

In collaboration with the content strategist, a content engineer does the following and more:

- Identifies how content varies based on customer needs and where each need arises in the customer journey
- Identifies how content can be modularized so that it can be automatically reused (mixed and matched) to meet customer needs
- Develops format-free, structured content models so that content can be written in a consistent way and automatically published to any channel (mobile, Web, print, wearables)
- Defines the structure of the CMS repository so that it supports authoring and content retrieval
- Develops metadata to tag all content modules for personalized content retrieval
- Develops business rules to identify how content should be assembled automatically upon customer request
- Determines how to optimize the content creation process
- Determines how content has to be structured and tagged to support chatbots, natural language generation/automatic content generation, and AI

Content engineers figure out how to take the vision and make it a reality in an ever-changing technological landscape.

Content Engineer: A Little Background

As content got more complex, content strategists identified a gap in the way we design and support content. We could certainly rely on information technology teams to help, but while they typically had an understanding of enterprise tools, like document management systems and the Microsoft Office Suite, they had little understanding of content that wasn't housed in a document. There were content strategists who were highly skilled at customer needs and content design, information architects that understood how to design the interaction on websites, and user experience designers who could design Web-based—and later, mobile—experiences, but trying to go from the vision to working reality was hard and often failed.

Joe Gollner proposed the concept of "content engineering" with his article "Architecting Information and Engineering Content (Gollner 2010). I and my co-producer Scott Abel made it a theme of the Intelligent Content Conference in 2012 to discuss how the content engineer role could be better defined. Content Science and Kanban explored the role in "5 Reasons Content Strategy & Content Engineering Go Together" (Jones 2014). In 2016, I detailed the idea further in an article for the Content Marketing Institute, "Why You Need Two Types of Content Strategist" (Rockley 2016). Cruce Saunders, the guest editor for this issue, followed with "The Many Disciplines of Content Strategy" (2016).

The content engineer is a relatively new role, or at least a relatively new *recognized* role; however, it has caught on and been accepted very quickly. A search of LinkedIn finds many examples of people who describe themselves as content engineers and job sites listing jobs for content engineers.

Content Engineer Responsibilities

Sometimes, in smaller organizations, the role of the content strategist and content engineer are combined into one individual. Some technical communicators will look at the role of the content engineer and say, "I do that, but I'm a content strategist," and they'd be correct, because technical communicators have been designing and developing component-based, multichannel, reusable content for decades now. The role might be called many things or might be combined with other roles, but no matter what it's called, the responsibilities are the same.

Typical responsibilities for a content engineer include the design and development of:

- Content models
- Reuse strategy
- Taxonomy and metadata
- Governance
- Technology selection and configuration

Content Modeling

Content models formalize the structure of your content in guidelines, templates, and structured frameworks. Models identify and document the structure on which your content is based. Content engineers are responsible for designing content models that identify:

- Content structure at the content assembly, component, and element level
- Relationships between content
- Content attributes

Reuse Strategy

Reuse enables you to create custom and dynamic assemblies of content, as well as deliver content on any device or platform. The reuse strategy defines your granularity of reuse (size of reusable components) and the ways in which content will be reused. The content engineer is responsible for creating a reuse strategy that defines:

- The granularity of reuse (component, fragment, variable)
- Conditional logic associated with reuse
- Types of reuse (identical or derivative)
- Ownership and governance

Taxonomy and Metadata

A taxonomy is a hierarchical listing of topics or subject categories supported by metadata. A taxonomy enables you to intelligently store and retrieve your content based on a common vocabulary and shared metadata. The resulting taxonomy is both customer-facing—such as for use on your website and social media platforms—and internal facing for your content management system. The content engineer is responsible for defining:

- Taxonomy for organization, storage, retrieval, and delivery of content
- Controlled vocabulary
- Reuse classification hierarchy
- Content flows and relationships

And although content engineers are rarely responsible for customer facing SEO metadata, they should expect to work with the SEO specialists and provide insight that will enable them to do their job in a more efficient manner.

Workflow

Workflow defines how people and tasks interact to create, update, manage, deliver, retire, and archive content. Workflow moves content from task to task and optimizes the content lifecycle, ensuring the business rules specific to your organization are followed at every step. Workflow is used to manage the content lifecycle and can be used to manage the business rules for the interaction and delivery of content. The content engineer is responsible for defining workflow that includes:

- Roles and responsibilities
- Tasks, interactions, dependencies, and governance
- Business rules

Governance

Governance focuses on balancing control and creativity within the organization. The term "governance" encompasses many things, including business processes, content creation, and publication workflows, as well as taxonomies and metadata management. The content engineer is responsible for creating a governance strategy that controls: content models, content, reuse, metadata, and workflow.

Technology Selection and Configuration

Today's content is only possible with an underlying robust technology. The content engineer may be responsible for determining the technology or suite of technology that best supports the content and customer requirements and managing all aspects of reporting on the health and functionality of the technology and toolsets.

Although the content engineer may not be the primary resource for these aspects of the project, they should work with those who are, to ensure appropriate technology selection and management.

More than Technology

The role of a content engineer can be very technical, but it is important not to lose sight of people and relationships, too. In many corporate situations a content engineer, particularly a senior content engineer, is also responsible for:

- Collaborating with content strategists to understand the needs and vision
- Working with business stakeholders to support corporate goals
- Ensuring the efficiency of content creation, reuse, management, and publication/delivery
- Acting as a liaison between the business and IT to support a strong partnership and a common understanding of the content contributor needs, customer needs, and the underlying technology
- Designing and managing user acceptance testing
- Developing, and potentially managing, a governance steering committee

Becoming a Content Engineer

Like many new disciplines and the slow rate at which colleges and universities develop new courses, there are currently no programs in content engineering. However, this is a role you can grow into or define in your organization by taking courses in related disciplines and learning by doing. Some suggestions for getting started include the following.

- Get a solid understanding of content strategy, not just "front-end" but "back-end" as well.
- Take a course or attend conference workshops in developing a taxonomy.
- Learn to code, at least at the basic level, and learn one or more of CSS, Javascript, HTML, DITA, or Markdown.
- Learn about SEO and how best to support it.
- Take a course in Web development.
- Take a course in user experience design.
- Learn about usability testing.
- Learn the basics of database design.
- Learn about developing for mobile operating systems and apps.
- Take the opportunity to not just use a CMS or CCMS but understand how it works.
- Get as much exposure to technology as you can. Tinker!

The role of a content engineer is an exciting and challenging one. In many ways, it's a lot like a stage manager in a theater or the flight director at NASA. In each of these roles you're expected to have a broad understanding of many disciplines, the ability to understand problems, create solutions, and work together with others to implement them quickly and efficiently. As a content engineer you may not be at the front of the stage, but you will be in the center of the action.

ANN ROCKLEY (rockley@rockleygroup.com) *is CEO of The Rockley Group, Inc. She has an international reputation for developing intelligent content strategies. With more than 30 years' experience as a consultant, she has been instrumental in establishing the field in structured content strategy, content reuse, intelligent content strategies for multichannel delivery, and structured content management best practices. Known as the "mother" of content strategy, she introduced the concept of content strategy in her 2002 ground-breaking book,* Managing Enterprise Content: A Unified Content Strategy, *now in its second edition. Ann is the founder of the Intelligent Content Conference. Ann has a Master of Information Science from the University of Toronto and is a Fellow of the Society for Technical Communication. Ann has written multiple books, is a frequent contributor to trade and industry publications, and is a keynote speaker at numerous conferences in North America, Europe, and Asia-Pacific.*

REFERENCES

Gollner, Joe. "Architecting Information and Engineering Content." *The Content Philosopher.* 21 February 2010. *https://www.gollner.ca/2010/02/architecting-information-and-engineering-content.html.*

Jones, Colleen. "Oreos + Milk: 5 Reasons Content Strategy + Content Engineering Go Together." *Slideshare.* 28 February 2014. *https://www.slideshare.net/leenjones/oreos-milk-5-reasons-content-strategy-content-engineering-go-together.*

Rockley, Ann. "Why You Need Two Types of Content Strategist." *Content Marketing Institute.* 22 February 2016. *https://contentmarketinginstitute.com/2016/02/types-content-strategist/*

Saunders, Cruce. "The Many Disciplines of Content Strategy." *Medium.* 7 March 2016. *https://medium.com/@cruce/the-many-disciplines-of-content-strategy-fccd6323d454.*

Follow Your North Star to a Successful Content Engineering Team

By CAROLYN SWIFT-MUSCHOTT

Shutterstock/Yuriy Mazur

I MAINTAIN A VERY SIMPLE definition of content engineering: a content engineer ensures that content has the right metadata and is in the right format to support customer needs on their chosen platform.

In reality, content engineering is often anything but simple. How do you know when you've reached the point where you need a formal content engineering team? What does that even look like? And how do you build an organization that will ensure successful delivery of content to the customer for every project? Here are some lessons I've learned along the way.

Recognize Your Tipping Point

Content engineers are responsible for making content "work" in digital—and print—channels. This job is often handled informally in smaller organizations, when working with small volumes of content, or when content is only expected to perform well on a single platform. In these cases, some responsibility for content engineering is often naturally assumed by business analysts, software developers, systems architects, or content strategists. As an organization grows or its needs change, you might start to recognize that you have a content engineering gap from the following signs.

- Content delivery meets editorial standards but requires frequent manual tweaking to get it to present well on your platform(s). Fixing one problem often creates another.
- There is constant friction between the content strategy team and software developers.
- Content presents well on your native platform but displays or behaves poorly on other platforms.
- The file system or content management system you're using does not allow you to easily find, edit, and version content for updating or reuse. You use file-naming conventions and folder structures to know what content was used where, when, and for what purpose. It's often difficult to determine what version of content was last published and for which purpose.
- The content team spends more time managing their content than creating it.
- A project has failed or resulted in sub-optimal outcomes, because the content delivered didn't integrate well with the application (or viewed another way, the application didn't support the content requirements).

Know What Skills to Hire

If these symptoms are familiar to you, you might consider adding a content engineering role to your organization. Here are a few skills to look for during the hiring process.

Technical Skills Are Important

Content engineering is inherently technical, so there are some technical skills that a content engineer can't live without. At minimum, content engineers typically need to have experience with relational databases, one or more query languages, Boolean logic, a thorough understanding of search and retrieval concepts, common structured data formats (XML, JSON, XHTML, etc.), and presentation markup. Depending on the organization and the roles of other project team members, they might also need to have a background in user experience or systems analysis, or experience with a specific programming or query language, data science, artificial intelligence, or machine learning. Certain types of content might require specialized skills, such as experience with file formats, closed captioning, and transcription standards for video content.

Technical Aptitude – Soft Skills = Ineptitude

Content engineering is all about building bridges between content strategists, product management, and technology teams. Technical skills are not worth much unless they come alongside excellent communication skills, an empathetic and curious mind, and the ability to influence and lead a team with a healthy balance of confidence and humility.

Why are these soft skills so critical for a content engineer? Consider the following scenarios.

- Presenting a business case to executive leadership to adopt a new CMS. The executive team is inclined to see technical complexity and expense. A skilled content engineer can tell the value story and demonstrate the connection to the bottom line.
- Facilitating discussions between content strategy teams and software engineering teams. The software team may roll their eyes at the outdated workflow the content strategy team uses to publish content, while the content strategy team pushes back on changes to their workflow. An experienced content engineer will empathize that the teams have unique needs and will work toward a solution that addresses those needs while keeping everyone focused on their common larger goal.

The User Is Your North Star

A good content engineer will keep in mind what user needs the business is working to support with each feature before they try to address how to accomplish it. A great content engineer will also ask why the user needs to do something.

Let's say, for example, that a product manager for a home furnishings e-commerce site asks for a feature to limit products by color. He points out that color for each product is already stored in the marketing CMS. He provides a wireframe showing the new limiter that will expand to show a list of colors and that a user can select one or more colors.

The outcome and the product manager's experience will likely be very different if he works with a team without a content engineer vs. working with an experienced content engineer.

Experience Without a Content Engineer

Product manager: Wait, this isn't what I wanted. The limiter includes hundreds of colors, many of which are

similar, and no user is going to scroll through that many colors. It's useless.

Tech lead: Well, that's the data you told us to use.

Product manager: I'm not going to approve the feature as-is. We need a more manageable list of colors.

Tech lead: I guess we'll have to refactor the feature. We could map individual colors to color families, but you'll need to provide the mapping. It's going to cost additional dev time and will delay the release by two weeks.

Experience with a Proficient Content Engineer

Content engineer: The sample data the software engineering team is using in their development environment is from sample records from one product type, which includes a range of 12 colors. I've done some analysis on the product database, and there are well over 100 unique color descriptors across all products. Once the application is pointed to the production database, the list of color options presented in the color limiter may be overwhelming for users.

Product manager: You're right, that won't work. We're going to have to do something to roll those up.

Content engineer: I can work with the marketing team to create a standardized vocabulary of color families and map the product colors to those families. Then we can update the database to include a color family field and add the color family to each record. I'll notify the software team that we want them to use that field for the limiter rather than the product color field.

Product manager: Perfect. Thanks!

Experience with a Great Content Engineer

Content engineer: Why do the users want to limit by color?

Product manager: They want to pick their favorite color; they want something that complements other items they have, that sort of thing.

Content engineer: Tell me more about how a user chooses something that complements other items they have.

Product manager: Once a user limits to products in a color family, they need to rely on photos to verify the color. Unfortunately, online photos are terribly unreliable sources, because each user's screen quality and settings vary, so this is very unreliable.

Content engineer: Would it help if we apply a standard color rating system, like Pantone, so that the color name means the same thing to every user?

Product manager: Well, going back to capture that information for existing products would be prohibitively expensive. The colors of the hard goods sold on the site are inspired by our company's paint colors, but again, our products aren't tagged with that information.

Content engineer: If we could find a way to map product colors to paint color family, would that be useful?

Product manager: Sure it would—it would ensure that the items customers are selecting are color coordinated.

Content engineer: I'll talk with the design team to see if we can tap into their design database to develop a mapping from product ID to paint color family in the e-commerce platform.

Product manager: This will enhance the user experience, and I'll bet we see an increase in sales, because customers will like the reassurance of shopping on a site where products are guaranteed to coordinate.

The lesson: don't dive so quickly into implementation that you lose sight of what you're trying to help your user achieve. Be more interested in your user than yourself and be inquisitive enough to really understand the user's needs.

Keep It Simple, Stupid (KISS)

When considering who you want on your team, look for people who seek the simplest way to solve a problem. This means focusing on the current problem at hand and the known variances rather than accounting for every scenario imaginable. The KISS principle is not, however, an excuse to ignore the need for extensibility. In fact, it makes extensibility more important, not less.

Imagine you're tasked with developing an XML schema that includes support for providing image metadata (title, image size, image format, etc.), and you create the XML structure shown in Figure 1.

```
<image imageWidth="640" imageHeight="480">
       <imageTitle>Content Engineering</imageTitle>
       <imageCaption>A content engineer demonstrates
how to build flexible workflows</imageCaption>
       <imageFormat>JPEG</imageFormat>
</image>
```

Figure 1. XML schema for image metadata.

Yet you can envision a day on which your organization might want to use videos. Well, don't spin your wheels developing a schema that handles all of the additional metadata that might be required for video (such as duration, transcript, etc.), but do consider that your schema is very clearly image-focused. Can you make it more generic? Certainly—for example, as show in Figure 2. It's still simple, just more extensible.

```
<media mediaType ="video" mediaWidth="1440"
mediaHeight="1080" mediaDuration="1:05:10">
              <mediaTitle>Content Engineering</mediaTitle>
              <mediaCaption>A content engineer demonstrates
how to build flexible workflows</mediaCaption>
              <mediaFormat>MP4</mediaFormat>
</media>
```

Figure 2. Still simple, but more extensible, XML schema for media metadata.

Engineer Your Team as You Would Your Content

Well-engineered content is standard enough to be used programmatically but flexible enough to be used in multiple ways. Likewise, a high-functioning content engineering team should have a standard, but flexible, tool kit that can be applied to any project. A content engineer's tool kit includes:

- Templates that inventory the envisioned content, map metadata to requirements, outline content workflows, and define data models;
- A problem-solving methodology that applies at the project level and at the individual requirement level—such as one of numerous problem-solving models: McKinsey, PDCA, DMAIC, 8D, etc.; the important thing isn't which model you use, but rather that your team has a standard approach to problem solving that can be applied in any context; and
- Structured analysis capabilities that enable a content engineer to propose, define, and explain data structure, flow, and data-system interactions—in particular, content engineers will be most successful when literate in data dictionaries, schemas, entity relationship diagrams, data flow diagrams, and system context.

Further, the team structure itself needs to be flexible. In a small organization, a single content engineer might be responsible for the entire content engineering effort. In a large organization, a content engineer might define the "what" and "how" but rely on many others—software engineers, taxonomists, quality assurance engineers, data scientists, content architects, etc.—to do much of the implementation.

There is no one-size-fits-all model for building a content engineering team. As soon as you optimize your team for your organization, you will find that technologies and business needs have changed, and you're already looking to refine your team in terms of roles and skill set. You might have relied heavily on manual tagging of content yesterday, but you might now find that you're looking to rely on machine learning to accomplish some of that work, which requires a new skill set.

Think about building and growing a content engineering organization as a wheel with spokes. At the center is the content engineer role. If the team is managed in a way that encourages continuous learning and an open dialogue about the team's constraints and limitations, the content engineers will be the key to identifying the level of

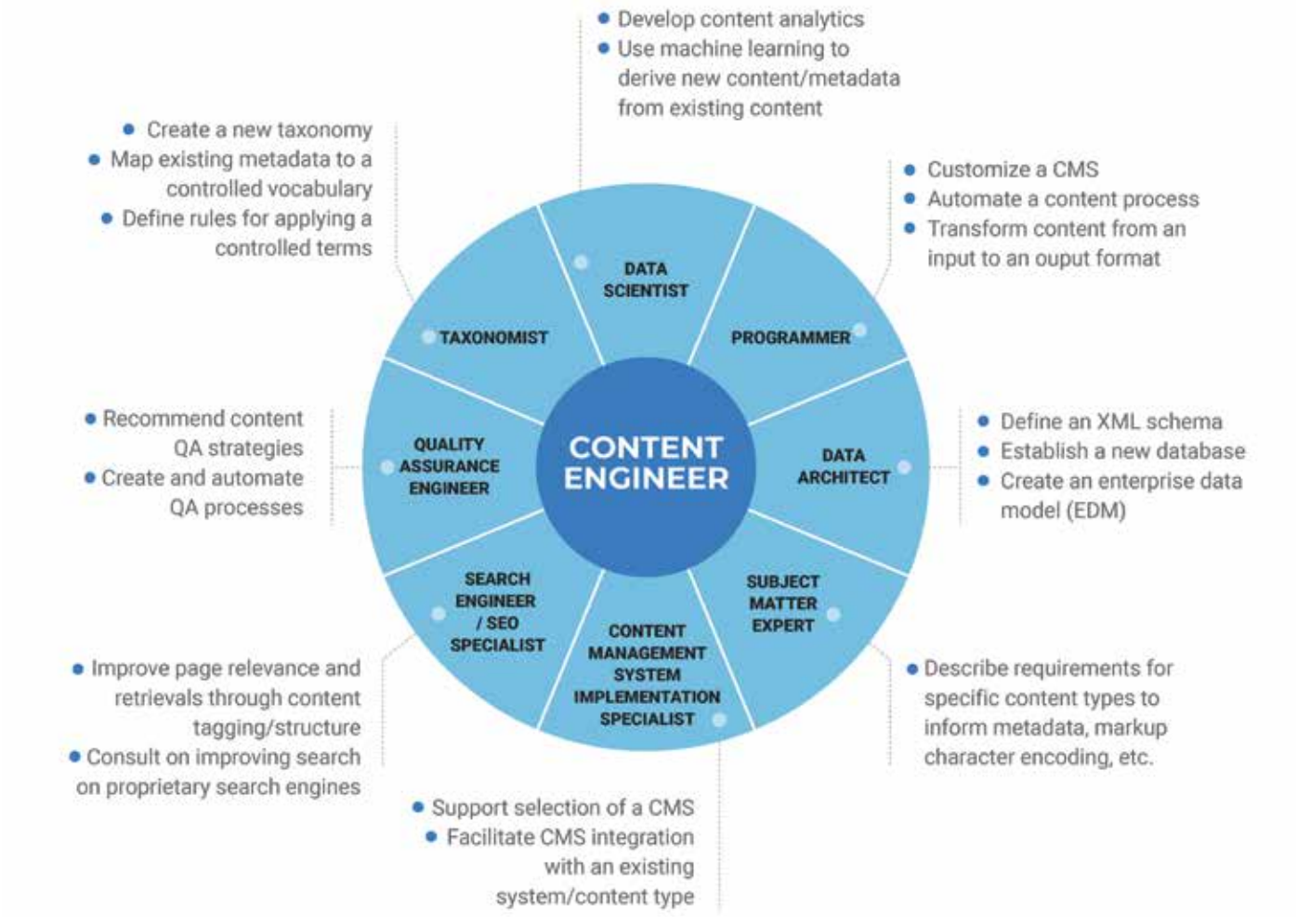

Develop content analytics
Use machine learning to derive new content/metadata from existing content
Create a new taxonomy
Map existing metadata to a controlled vocabulary
Define rules for applying a controlled terms
Customize a CMS
Automate a content process
Transform content from an input to an ouput format
DATA SCIENTIST
TAXONOMIST
PROGRAMMER
CONTENT ENGINEER
QUALITY ASSURANCE ENGINEER
DATA ARCHITECT
Recommend content QA strategies
Create and automate QA processes
Define an XML schema
Establish a new database
Create an enterprise data model (EDM)
SEARCH ENGINEER / SEO SPECIALIST
CONTENT MANAGEMENT SYSTEM IMPLEMENTATION SPECIALIST
SUBJECT MATTER EXPERT
Improve page relevance and retrievals through content tagging/structure
Consult on improving search on proprietary search engines
Describe requirements for specific content types to inform metadata, markup character encoding, etc.
Support selection of a CMS
Facilitate CMS integration with an existing system/content type

Figure 3. Content engineering principles at the center of an organization.

content support your organization needs. They will know if they need programmers to automate content processes, they'll recognize when taxonomy requirements have become too complex, and they'll identify areas where machine learning expertise will provide significant lift. In this way, content engineering principles are at the center of the organization (see Figure 3), but the team may grow to include members from other disciplines.

A Good Sketch Is Better than a Long Speech

Content engineers should be very comfortable whiteboarding and diagramming. An effective picture can reduce a conversation from hours to minutes and can quickly create alignment across multiple functions. For example, compare the readability of the "written database migration plan" in Figure 4 with that of the "visual database migration plan" in Figure 5.

Get Your Head Out of Your ASCII

Content engineers need to know how to sell the value of their work—and the way to do that is rarely by throwing around technical jargon.

For those who don't work in content disciplines, the concept of content engineering can seem vague, but executive leadership is not going to approve a new position or fund a content engineering proposal if they don't understand the value it provides. Further, other teams are not going to engage with yours if they don't know what you bring to the table.

It is critically important that content engineering leadership and their staff always relate their work to its business value. Prefer statements like, "We're enhancing our CMS to enable our content developers to create more relevant content faster and drive more traffic to our site through links to related articles, with an anticipated result of a 15 percent increase in qualified leads," over statements such as, "We are enhancing our CMS so that it can handle more scientific character entities and assigning metadata to our articles based on a new topic taxonomy." They may both be true and accurate, but the first statement is readily understandable by a wider audience, and it demonstrates why the effort is important to the business.

Plan for Migration to Sparkly Database

The Sparkly DB was established as the content store of record for all content created from 1/1/2018 onward, with plans to retire the Retro and Vintage article databases.

Retro and Vintage were used by different editorial teams who largely authored their own content and maintained discreet content databases. There were times when the two teams shared content; in order to facilitate deployment to the appropriate production environment, the teams would copy the shared content into their own database, modify it, and release it to their specific web site. This created articles that were largely duplicates of each other. Both teams have been using Sparkly DB since 1/1/2018, but have to go into Retro or Vintage whenever they want to update an existing article from dates prior. The goal is to get all recent content into a single database to streamline the editorial team's work and reduce maintenance costs.

The buckets of content involved in this migration are as follows:

Retro DB includes content from 2005 - 12/31/2017. The plan is to archive all content with a publication date prior to 1/1/2009. A subset of this content represents articles that are duplicated in Vintage DB.

Vintage DB includes content from 2007 - 12/31/2017. The plan is to archive all content with a publication date prior to 1/1/2009. A subset of this content represents articles that are duplicated in Retro DB.

Sparkly DB includes content from 1/1/2018 to present. The plan is to retain all of that content, and add Retro DB and Vintage DB content from dates 1/1/2009 - 12/31/2017.

The editorial team that historically used the Vintage DB (and now uses Sparkly) had a content production issue that caused some of their articles to be created without having an article type assigned. Article type is used to map articles to the correct area of the organization's web site, so article type will be assigne to those missing articles during migration. This production issue continued into the timeframe that the team was using Sparkly, so a subset of the Sparkly articles will also need to be retrofitted with article type. Article type will be assigned to the Vintage and Sparkly articles with null article type using a rules-based script.

Figure 4. A written database migration plan.

Content engineers should be very comfortable whiteboarding and diagramming. An effective picture can reduce a conversation from hours to minutes and can quickly create alignment across multiple functions.

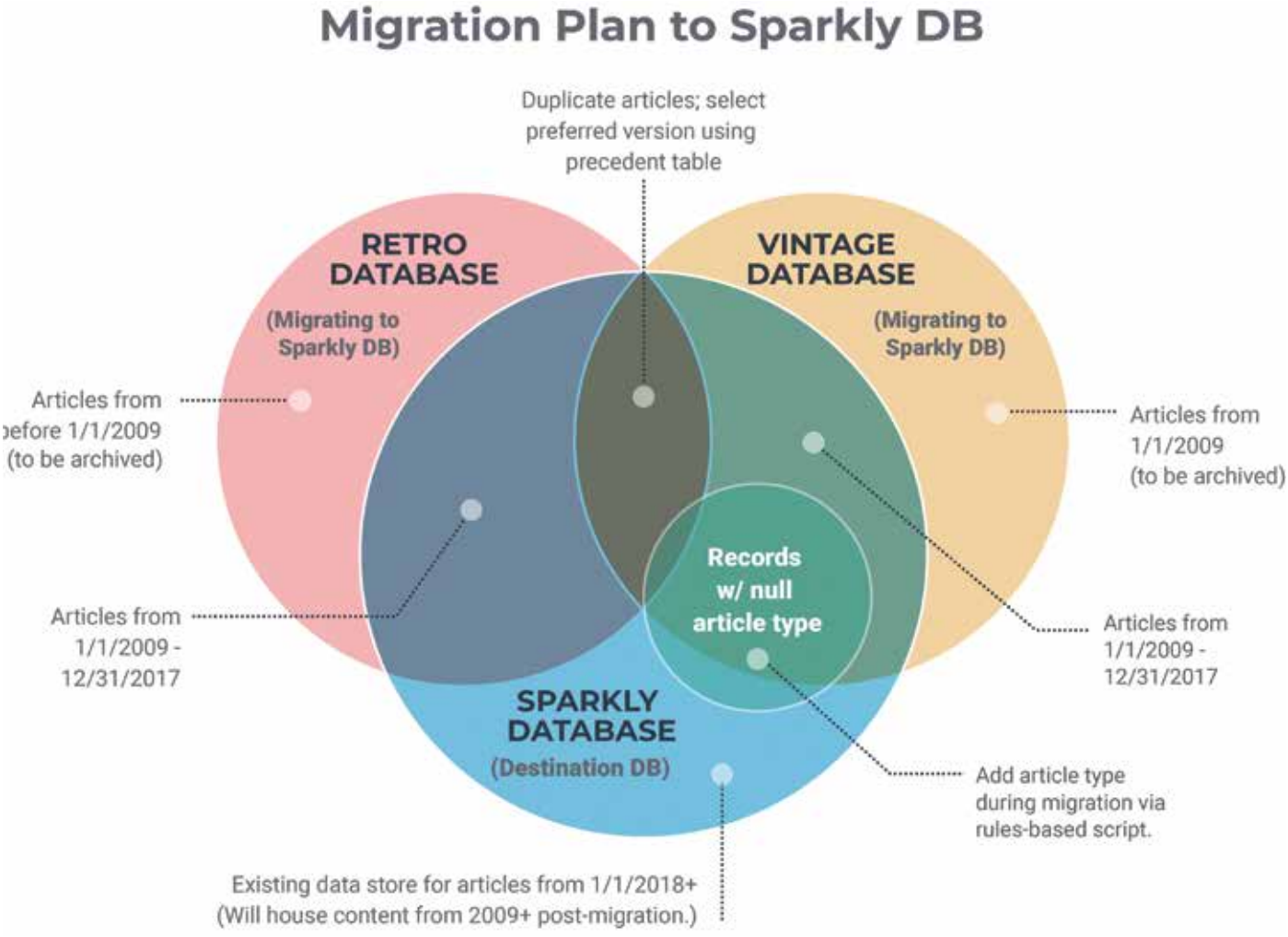
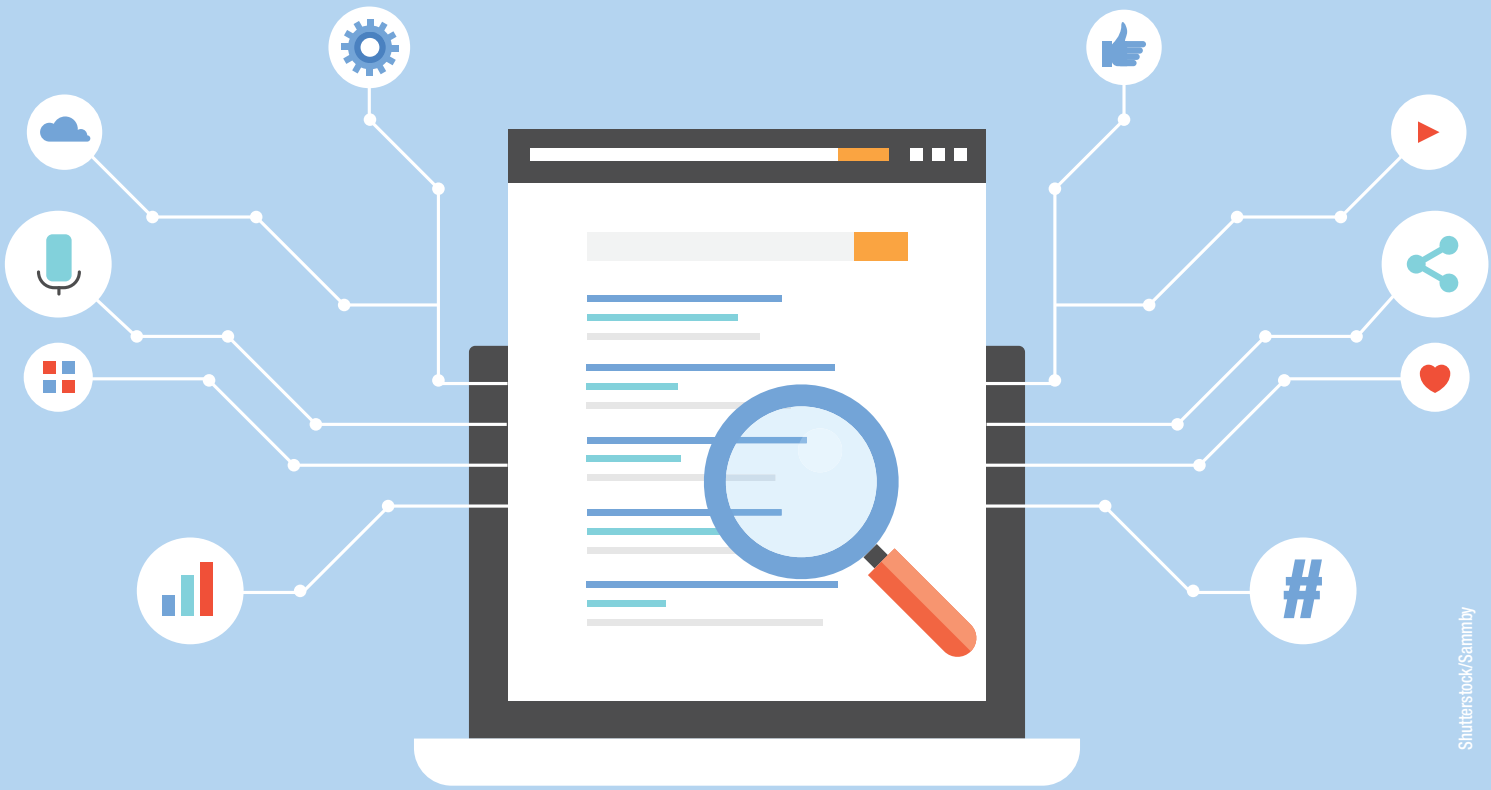

Migration Plan to Sparkly DB
Duplicate articles; select preferred version using precedent table
RETRO DATABASE
(Migrating to Sparkly DB)
VINTAGE DATABASE
(Migrating to Sparkly DB)
Articles from before 1/1/2009 (to be archived)
Articles from 1/1/2009 (to be archived)
Articles from 1/1/2009 - 12/31/2017
Records w/ null article type
Articles from 1/1/2009 - 12/31/2017
SPARKLY DATABASE
(Destination DB)
Add article type during migration via rules-based script.
Existing data store for articles from 1/1/2018+
(Will house content from 2009+ post-migration.)

Figure 5. A visual database migration plan.

Putting It All Together

A content engineering team can be a powerful asset to any organization when it reaches a tipping point in the complexity of its content-system interactions. As it may be a new concept to your organization, it is important to socialize the intent of the team, get buy-in from the other teams with whom content engineering would interact, and base the team's practices on solid engineering principles with an underpinning of strong technical and soft skills. Most importantly, never lose sight of your north star: the user.

CAROLYN SWIFT-MUSCHOTT *is Director, Content Engineering at Gale, a Cengage company in Farmington Hills, MI. She is privileged to work with a great team of content engineers, analysts, architects, and software engineers who cultivate the company's content and metadata strategy and develop the workflows and systems that support them. Her favorite aspect of content engineering is the collaboration and knowledge sharing across disciplines. You can connect with her on LinkedIn:* https://www.linkedin.com/in/carolyn-swift-muschott-3a80a317/.



Shutterstock/Sammby

Engineering Content for Superior Search Performance: Introducing Structured Data

By AARON BRADLEY

Machine-Driven Search Matters More than Ever

Customer-facing content sets, once hidden deep on websites and locked into print and PDFs, are now instantly searchable and discoverable. Helping Google and Bing—and our internal search engines—to find and deliver relevant content has become a key role of publishers across marketing and technical documentation. Engineering content for discovery takes some work but delivers significant returns through increases in content consumption.

Historical Perspectives on Structured Data

One of the primary ways in which digitally rendered content on the Web differs from its printed equivalent is the ability of machines to ingest, analyze, and index that Web-based content. This in turn allows machines to return relevant content items in response to a search query or to otherwise surface interesting or useful content based on a user's interests and needs.

In their efforts to understand what any given piece of digitally provided content is about, enterprise search engines like Google and Bing are greatly aided by having that content made available to them as *structured data*. As the name suggests, structured data is content provided in a very specific format that the consumers of this content explicitly understand.

In the early days of the Web, the ability of search engines to identify the precise facts expressed on a Web page was only as good as their ability to parse *unstructured* content and transform it into consistently classified information. Regardless of how sophisticated the approaches employed, search engines were still ultimately guessing about the data they found on Web pages: three star icons encountered next to a restaurant review *probably* meant that the critic's rating was three, and that the rating was *probably* out of five, but there was no way that a search engine could know this for sure.

In an effort to reduce such ambiguity, in the late 2000s Google began to support HTML-based *structured data markup* that allowed Web publishers to provide very precise information about certain types of things, like reviews. Using the standardized vocabulary of microformats (a community initiative launched in 2005) or of *data-vocabulary.org* (a 2009 Google project), a Web publisher could now declare that the star icons on a page represented a rating and that the rating had a value of *exactly* three out of a possible score of *exactly* five.

While these efforts allowed Google, as shown in Figure 1, to produce the Web's first "rich snippets"—by which things like review scores could be displayed next to a page's snippet directly in the search results—effective structured data use was hampered both by the lack of a single standard for Google and the absence of any structured data standards whatsoever across search engines.

Salad - Thai Green Mango Salad Recipe
5 reviews - Total cook time: 20 mins
You asked for a one-page printable version of my step-by-step Green Mango Salad recipe, so here it is! This salad will blow you away with its ...
thaifood.about.com/od/thaisnacks/r/greenmangosalad.htm -
Cached - Similar

Figure 1. The example accompanying Google's announcement of recipe rich snippets in April 2010. At this time the snippet was generated based on the microformat hRecipe, or from the Recipe item type at data-vocabulary.org, *a predecessor of* schema.org.

Introducing Schema.org

In June 2011, Google, Bing, and Yahoo! addressed the gap in structured data markup standards head-on by jointly announcing the availability of *schema.org*, a common set of standards that, in Google's words, "aims to be a one stop resource for webmasters looking to add markup to their pages to help search engines better understand their websites" (Goel and Gupta 2011). Russia's largest search engine, Yandex, signed on to the initiative later in the year.

These standards provided a common set of terms for publishers to describe things present in their Web content (vocabulary) and approved methods of encoding this information for search engine consumption (syntax).

This degree of cooperation between search engines is rare (the only notable prior example being a 2006 agreement between Google, Microsoft, and Yahoo! on a protocol for XML sitemaps), and the impact of their collaboration around *schema.org* on the adoption and utility of structured data cannot be overstated.

For search engines and publishers alike, alignment on a single vocabulary has made it easier to refine and grow that vocabulary. More so than previous efforts, *schema.org* is a living standard, and it has become more expressive over time to satisfy the demands of well-articulated use cases. Launched with just under 300 types (the things the vocabulary allows publishers to describe, like events or products or videos), *schema.org* today boasts more than 1,100 types. Table 1 lists the most commonly-used *schema.org* types eligible for Google rich results (Web Data Commons 2018).

Table 1. The most commonly-used schema.org types eligible for Google rich results.

Type	Number of Hosts
schema.org/Organization	1859502
schema.org/LocalBusiness	543356
schema.org/Event	151729
schema.org/Review	124022
schema.org/Product	121332
schema.org/Recipe	21195
schema.org/Restaurant	10854
schema.org/Book	10587
schema.org/Movie	9194
schema.org/JobPosting	9151

schema.org has also evolved a better framework for community engagement and vocabulary-building than prior standards development efforts. While the *schema.org* Steering Committee, which is representatively still search engine-heavy, has ultimate control over which new vocabulary is added to the schemas, non-search engine participation in the Steering Committee is now formalized, and there are multiple avenues for interested parties to participate in vocabulary development.

Search engine alignment also makes it more likely that Web publishers will go to the trouble and expense of providing structured data markup, both because the benefits of doing so are not restricted to a single search engine's results pages and because the search engines' commitment make it more likely that these benefits will be enduring rather than fleeting. And those benefits, both for end users and Web publishers, are not insubstantial.

Superior Visibility in the Search Results

The enduring value proposition for using schema.org is the generation of enriched search results, also known as "rich snippets" or simply "rich results." These are visually distinct search results that prominently display important schema.org-encoded values for specific types of content. For example, a rich result for a recipe might display the recipe's ingredients, preparation time, calories, and review ratings (see Figure 2).

Baked Ziti I Recipe - Allrecipes.com
https://www.allrecipes.com › recipe › baked-ziti-i
Rating: 4.7 - 5,815 reviews - 55 min. - Calories: 578
Directions. Bring a large pot of lightly salted water to a boil. Add ziti pasta, and cook until al dente, about 8 minutes; drain. In a large skillet, brown onion and ground beef over medium heat. Preheat the oven to 350 degrees F (175 degrees C). Bake for 30 minutes in the preheated oven, or until cheeses are melted.
Generated from schema.org/AggregateRating
aggregateRating
type AggregateRating
ratingValue 4.70
reviewCount 5815
Generated from schema.org/Recipe
totalTime PT55M
recipeInstructions Bring a large pot of lightly salted water to a boil. Add ziti pasta, and cook until al dente, about 8 minutes; drain. In a large skillet, brown onion and ground beef over medium heat. Add spaghetti sauce, and simmer 15 minutes. Preheat the oven to 350 degrees F (175 degrees C). Butter a 9x13 inch baking dish. Layer as follows: 1/2 of the ziti, Provolone cheese, sour cream, 1/2 sauce mixture, remaining ziti, mozzarella cheese and remaining sauce mixture. Top with grated Parmesan cheese. Bake for 30 minutes in the preheated oven, or until cheeses are melted.
nutrition
type NutritionInformation
calories 578 calories;

Figure 2. A rich result for a recipe in Google, with data sources illustrated.

For eligible result types, this can lead to a substantially better experience for search engine users, especially when those users are searching on a mobile device. The presence of rich results makes it much easier for a user to assess the potential usefulness of a Web page without having to visit it. For example, a user might be able to avoid clicks on pages for past events in a results page of events listings, or to skim recipes to focus on those under a certain preparation time, or to explore products only within a certain price range.

This benefit is extended to Web publishers as well, insofar as those search engine users are more likely to be consuming only relevant content from a publisher, allowing them to avoid the aggravation of purposeless visits (and the negative association with the brand in question). In situations where one publisher's offerings are similar to another's, both the visually distinct result and the information provided within it make it more likely that a search user will click on a rich result than on the plain "blue link" of a Web page that lacks structured data markup.

Depending on the search engine, structured data might be required to generate rich results *and* to include those same pages in search verticals that are a subset of the full search results. For example, as shown in Figure 3, job posting markup fuels the display of a block of job rich results in Google with a "more jobs" link that clicks through to a set of result pages that display job postings exclusively. A job posting from a site without *schema.org/JobPosting* markup *might* appear in the Web results for a relevant query but *will not* appear on the screen after a user clicks on "more jobs."

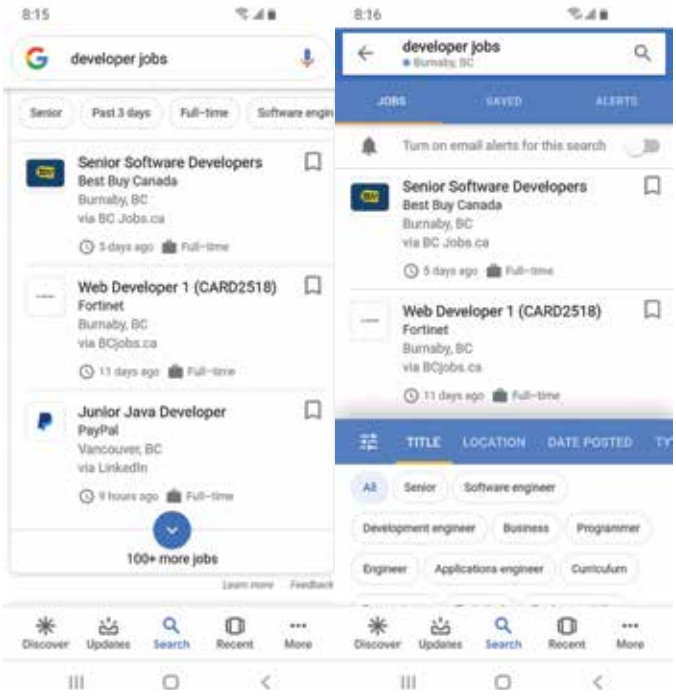


Figure 3. Web pages with schema.org/JobPosting *markup are eligible for rich results in Google (left), and appear in a dedicated job search vertical when a search user clicks through on "more jobs."*

Increasingly, *schema.org* data is also being used as a mechanism to enrich the search engines' knowledge graphs. This potentially extends the utility of a piece of content from a listing in a search engine's document index (those "ten blue links") to a presence in that engine's knowledge base of facts about things, with structured data-provided facts surfacing in features such as Google's Knowledge Panels or in search-engine voice responses.

Search Engine Discoverability

Structured data markup makes it much easier for search engines to understand a Web page—and in particular the entities and information about those entities to which a piece of content makes reference. While structured data use alone does not provide Web pages with a boost in search engine rankings, the search engines' superior understand-

ing of pages with structured data makes it more likely that these pages will appear in results for relevant search queries.

schema.org, for example, makes it easy for publishers to provide data that allows a search engine to determine to which one of two or more similar entities a piece of content refers, such as to which of the four Wisconsin towns named "Springfield" an article is referring or which of several similar products is being described on a given product detail page.

Search engines are increasingly leveraging structured data as a means of discovery for content that cannot otherwise readily be parsed to uncover meaning. For example, Google is using schema.org and Dataset markup to ingest publisher-provided information about datasets whose meaning would otherwise be opaque to the search engines and to provide searchers with results about the content of these datasets.

Similarly, structured data allows publishers to provide information that can be used by search engines to generate interactive experiences within search results. A search result for a song, for example, might include links powered by structured data that launch that song directly in a streaming service.

The better visibility content has in search engines, and the more structured-data links included in the results, the more likely that content will come to the public's attention.

Search Engine Transformation

The power of structured data to fuel things like dataset discovery and media actions is largely predicated on the separation of a Web page's presentation layer (what a user sees) from its data layer (what a search engine consumes).

The provision of this data layer is a de facto method of generating intelligent content—content that is "structurally rich and semantically aware"—which in turn allows this content to be reconfigured and reused for additional publishing endpoints.

Search engines are now starting to transform Web page-provided content to other endpoints to surface this content outside of Web search results. For example, a recipe published using Google-prescribed *schema.org* Recipe markup is automatically eligible to be returned as audio on a Google Home. Similarly, *schema.org's* speakable property allows publishers to identify content that's "especially appropriate for text-to-speech conversion" for search engines to preferentially return this content in response to a voice search.

Just as *schema.org* has supported improved user experiences on mobile devices by reducing a searcher's need to browse websites, so it now supports the easy ingestion of Web-provided information on smart speakers and smart displays. Because *schema.org* is endpoint agnostic, it is a method by which publishers can future-proof their content, giving that content at least a fighting chance to appear in future search results and on future devices.

Figure 4. Web documents annotated with schema.org *structured data may be returned as search results in numerous Google endpoints aside from their standard "10 blue links."*

Conclusion

Structured data is a method of providing precise, machine-readable information about the structure and meaning of a piece of content.

Providing structured data might improve the discoverability of content by search engines and result in higher visibility in search results, as well as potentially make Web page-based content available on other devices.

Fundamentally, structured data liberates the meaning of content from its visual presentation, making it easier for search engines to understand, use, and transform this content.

All of the parties involved in the production and use of *schema.org* are potential beneficiaries of structured data. Search engines benefit by better understanding the content they're indexing and by being able to build new search products based on prescribed schema use. Publishers benefit by the additional exposure their content receives in the search results—both in terms of visibility and relevancy—and by the potential of that content to now reach more users on a greater variety of devices.

Engineering content for search performance starts with getting familiar with *schema.org* and then generating your own markup. Whether from scratch, by using generators, or with the help of developers, adding structured data to content should be on the agenda of any publisher interested in content discovery.

AARON BRADLEY (aaranged@gmail.com) *works as a Knowledge Graph Strategist at Electronic Arts in Vancouver, Canada. As a champion of intelligent content, his days are consumed with the ontologies, taxonomies, and content models that bring a content graph to life. He can be found tweeting at* https://twitter.com/aaranged.

REFERENCES

Goel, Kavi, and Pravir Gupta. "Introducing Schema.org: Search Engines Come Together for a Richer Web." *Google Webmaster Central Blog.* 2 June 2011. *https://webmasters.googleblog.com/2011/06/introducing-schemaorg-search-engines.html.*

"Class-Specific Subsets of the Schema.org Data contained in the November 2018 Web Data Commons Corpus." *Web Data Commons.* 12 December 2018. *http://webdatacommons.org/structureddata/2018-12/stats/schema_org_subsets.html.*

Shutterstock/Vladyslav Starozhylov

Want Personalization?

Start with Content Engineering

By LISA TRAGER

THERE'S A LOT of talk about personalization these days. What most of these conversations skip are the foundational content engineering elements that will enable the content to be personalized.

Over the past 20 years working with enterprise content within large agencies, as a solo consultant, and deep in the trenches of Fortune 50 companies, I am now convinced that the only viable path to personalization involves building a strong practice of content engineering. Here's how my thinking has evolved.

Discovering the Magic of XML for Personalization

I'll never forget the first time I was exposed to XML. In the late 1990s, I worked as an Information Architect and Content Analyst for an agency that specialized in intranets for Fortune 500 companies. One project was related to creating a good customer experience for employees who used the intranet for HR-related information. With the magic of XML, we could provide personalized information about health benefits for over 50,000 employees based on their role, department, and location.

It's memorable, because at the time, it was so new, and the potential was mind-blowing.

While working at top New York agencies, I worked as a Content Strategist—a role that was mostly limited to thinking about marketing content for large corporate clients. Along with the team of content strategists I managed, we spent much of our time in spreadsheets auditing content and creating content matrices. I remember trying to introduce the idea of creating a content model, tagging for personalization, and structuring the content for reuse. No way. Applying content engineering best practices in an agency environment was unwelcome and, frankly, not understood. We were tasked to only "do content strategy." Doing anything besides marketing and copywriting was out of scope and certainly not included in the client's budget. Even when my suggestions would have ultimately saved the client time, resources, and budget, there wasn't even an opportunity to pitch it. I finally decided to accept a full-time opportunity on the client side—a chance, I thought, to make a bigger difference.

Working Inside a Giant Content-Driven Enterprise

Fast forward 20 years after my first exposure to the personalization offered by XML—this time working on staff at a complex 100,000 person company. I was recruited to be the "change-maker" who could come up with solving how to manage over 40,000 pages of support content. Front and center: the challenge of fixing a myriad of content problems. And, wow, the problems were huge.

Challenges included:

- **Content issues:** Poor formatting, repetitiveness, limited personalization, an ever-growing amount of content.
- **Technical issues**: Two separate, incompatible content management systems; lack of structure; lack of tagging; content not optimized for search; inability to publish relevant content across platforms (omnichannel publishing) to the Web, app, and chatbot.
- **Organizational issues**: Separate departments that "own" the content, territorialism, lack of will or interest in seeking viable solutions that go beyond "business as usual," fear of change that might bring risk to their annual performance review and associated bonus.

A viable solution that would have taken the organization from crawling to running within two years was within reach, and it included everything from creating a taxonomy to consideration of adding a CaaS platform. For this company and other large organizations, the investment needed to put a working content engineering plan with content reuse in place would have been negligible compared to the long-term costs of creating and maintaining content in a manually driven system.

The Value of Content Reuse

In addition to the hard and soft savings in time and resources, implementation of foundational elements, new processes, and approaches bring an improved customer experience, increase Net Promoter Scores (NPS), and prevent churn, which are worth gold in today's customer centric business world. According to a 2018 Forrester report,

> *The average Fortune 500 company is spending millions of dollars in waste on content operations: Teams are weighed down with inefficient workflows, manual workarounds are still failing to deliver a next generation customer experience, and things continue to get worse as content volume expands* (Forrester Consulting 2018).

In Mark Lewis' book *DITA Metrics 101* (Lewis 2012), he lays out cost models based on comparing various scenarios from using simple topics to more complex projects that use filtering, structured authoring, content reuse, and multiple translations. By defining the cost basis for each unit and then comparing costs of traditional publishing vs. with DITA or content reuse, the models prove the significant savings possible.

> *Because so many processes are tied to other processes, the savings can have a compound effect. For example, content reuse during the content development process results in savings that cascade in the publishing and translation processes* (Lewis 2012).

Building Blocks

Based on my experience working in an assortment of environments, the following are the ingredients that will help any organization build a strong foundation for managing complex and ever-growing content challenges.

Shared Taxonomy

Why?

Having an official taxonomy helps everyone from writers to IT understand the hierarchy of topics. The nodes would include not only information for sales related to products and accessories, but can also be used to inform the organization of related support and service content.

A shared taxonomy is the only way we can share the relevant tasks, product hierarchy, features, time, place in the user journey, and all the other facets that align content with specific user intents.

How?

You build a taxonomy to support personalization use cases to help inform the relationship of topics to users and where that content might be most useful in the customer journey. Using an example for child nutrition based on a "mad-libs" approach, the taxonomy might include breaking down moms, age of infant, product type; and product features: "As a new mom, I'm looking for the right formula to feed my newborn infant."

The best way to complete a complicated taxonomy for a brand with lots of products and use cases is to hire a

taxonomist. An expert with the experience to put into place a taxonomy that will have the biggest bang for the buck will not only save time but help avoid politics.

Building a shared taxonomy requires lots of input from various stakeholders and subject-matter experts. Don't try to boil the ocean. Think about what will have the most impact and return on investment (ROI). For example, in my last endeavor, I recommended that the taxonomy not include products, but relate more to the topics and terminology for service and support content.

I'm also a big fan of using software. For large sites, there is no way to govern or maintain a spreadsheet with hundreds of thousands of terms and ontological relationships.

Semantic Tagging

Why?

Tagging content based on meaning enables bots to better find and serve the right content to the right person at the right time. Semantic tagging enables the dynamic presentation of content that is defined by rules and logic.

How?

Semantic tagging hinges on the taxonomy you create, which defines the:

- Audience
- Topic
- Subtopic
- Intended use
- More

In addition, many people use schema.org for structured data available to search engines. Developed by Google, Microsoft, Yahoo, and Yandex, *schema.org* uses an open process based on community input to keep microtagging vocabularies relevant and to meet the needs of the wider developer community.

Content Reuse

Why?

As mentioned previously, the financial savings of content reuse includes the production costs of creating and maintaining the content across platforms and with each translation. The other benefit is that increased trust comes with consistency in messaging and information, regardless of where your customers are.

In particular, having a content reuse strategy in place is particularly relevant when it comes to support or technical content. When you think of support content for a telecom, how many ways can you give instructions to find Wi-Fi, add a contact to an address book, or even activate a device? With content reuse, rather than having hundreds of files for one topic, one can have one file with exceptions that address unique requirements, thereby making it easier to update and publish.

How?

Darwin Information Typing Architecture (DITA) is an XML data model designed for efficiency in writing and most importantly ease of reading instructional information. The major categories used to map the content include:

- Tasks: Used to describe how to perform a procedure
- Concepts: Descriptive information to help the user understand the background and context of a subject
- References: Topics that provide detailed facts

Structured Content

Why?

For omnichannel to work, having in place the technology, as well as strategy, to implement content reuse not only saves time and resources but helps ensure consistency. Clear structure provides the scaffolding and is directive for writers, giving them all of the advantages of form-based writing. Structure helps authors focus and cuts out anything not immediately relevant to the topic.

Structured content is future-proof content. As the market continues to rapidly create new unknown devices, platforms, and channels, the only way to enable content to remain accessible is by separating the message from the format. When embedded in HTML, content is locked into format.

How?

At the heart of structured content are components that can be reused across multiple platforms and/or channels. Through the use of a component content management system (CCMS), content is separate from format, enabling the content to look proper regardless of the output. Created as smaller chunks, these modules are also free from being tied to one long document, so the modules can be used individually or combined as needed to fit difference use cases.

Content-as-a-Service (CaaS)

Why?

Sometimes it just doesn't make sense to force a working group to change the system they are using for content creation. For example, in one case, AEM (Adobe Experience Manager) was the primary CMS for public facing content. Other departments, however, were using another CMS specific to the needs of internal agents and reps, which was not compatible with AEM. Each system had benefits, but neither system had content models or a consistent way that content itself was structured. In today's omnichannel world, having one content model (e.g., the [A] Master Content Model) applied across disparate content, makes it possible for systems to work together, as well as empowering multiple teams access to work on the content either together or independently. Making that unified content available via API enables us to personalize the content and to provide an enhanced customer experience.

How?

Although we hear a lot about "one source of truth," in reality that is usually not viable in large enterprises. That said, what is not only doable, but critical, is creating content models that can be shared across systems. Any headless platform, including AEM, that uses APIs can participate in a shared CaaS environment—the benefit being the seamless integration of content, which benefits workflow, content creation, and best of all, the ability to publish across platforms, channels, and print.

Content Operations and Governance

Why?

Content operations is an emerging function in corporate organizations and still rarely understood. The purpose is to have staff with subject matter expertise in all of the things related to the *production of content*, such as taxonomists, information architects, content strategists, and content engineers. As Rahel Bailie explains, the working definition of Content Ops is:

> *A set of principles that results in methodologies intended to optimize production of content, and allow organizations to scale their operations, while ensuring high quality in a continuous delivery pipeline, to allow for the leveraging of content as business assets to meet intended goals* (2018).

Another benefit of having a content ops group is that it can be the perfect vehicle for providing content governance, especially in large organizations, where it's often more like the wild west, and rules or guidelines are either unknown or not followed outside of that siloed unit.

Having a centralized group that can apply manual labor, automation, and software to ensure observance of guidelines has multiple benefits:

- Faster content creation and time to market
- Ability to create content at scale
- Improved quality
- Greater consistency in tone, voice, terminology, and information
- Reduced legal risks
- Greater adherence to branding and regulations
- Improved customer experience due to personalization strategy
- Automation and a move toward AI

How?

Develop a team with expertise in taxonomy, tagging, content reuse, DITA, and information architecture**.** Consider software like Acrolinx, which works like a virtual editor, applying the governance and rules defined by content ops to any writer in the company who has the software installed on their computer.

Content governance includes addressing everything from adding efficiencies to content workflows, to creating content strategy and editorial guidelines, taxonomies, and archiving standards.

Lessons Learned

The change plan is still very much a work in progress. Big changes in enterprise content lifecycles take big effort over long periods of time, and change is hard. What I experienced isn't that unique due to the realities of corporate culture. Until we can address both the rational and emotional reasons behind the resistance to change, the benefits that content engineering promises will remain an ideal discussed in conferences or written in books read by content engineering and content strategy insiders—and ignored by corporate decision-makers.

Companies working on transformation of content processes and systems need to address silos, territorialism, fear, lack of knowledge, constant reorganizations, and lack of accountability.

Without a commitment to making the necessary changes in process, roles, and technology, progress toward addressing ever-mounting content challenges will at best remain an uncertainty and at worst become a growing burden. Without a real change in the mindset of executives around prioritizing the importance of foundational changes and need for content engineering best-practices, creating ideal, personalized, customer experiences will continue to be a white whale.

LISA TRAGER *helps organizations develop and execute digital strategies for an omnichannel world. As a consultant and the Principal of Content in Motion, she led digital enterprise initiatives on the agency and client side in the telecommunications, healthcare, pharmaceutical, financial, consumer goods, and government sectors. Currently, she works at Verizon in Digital Operations where she applies her extensive knowledge and expertise in creating customer-centric experiences related to service and support.*

REFERENCES

"A Guide to Mastering the Master Content Model." [A]. October 2018. *https://simplea.com/Publications/Whitepapers/master-content-model.*

Bailie, Rahel. "Principles, Not Prescriptions: The Coming of Age of ContentOps." October 2018. *https://gathercontent.com/events/principles-not-prescriptions-the-coming-of-age-of-contentops.*

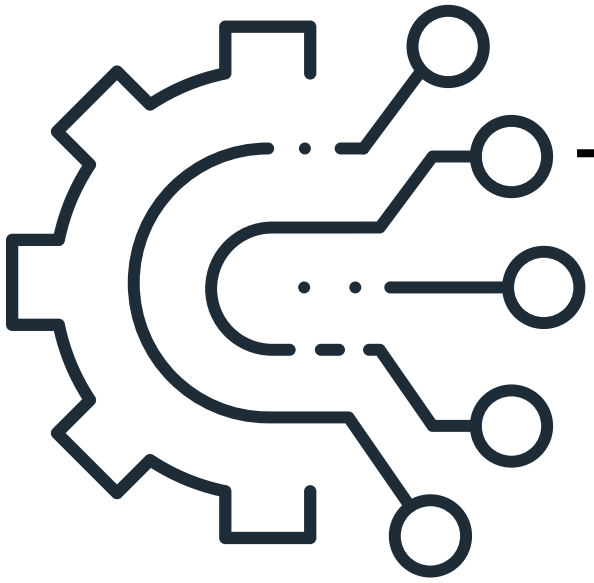
EasyDita by Jorsek. 31 July 2014. *https://easydita.com/when-to-use-the-topic-concept-and-task-types-in-dita/.*

Lewis, Mark. *DITA Metrics 101: The Business Case for XML and Intelligent Content.* The Rockley Group: Schomberg, ON, 2012.

Morse, Stephen. "When to use the Topic, Concept and Task types in DITA."

"Today's Content Supply Chains Prevent Continuous Customer Journeys." SDL. 11 November 2018. *https://www.sdl.com/resources/downloads/whitepapers/customer-journeys/whitepaper.html.*

The Roles of Standards and Governance in Engineering Content for Multichannel Distribution: Why Standards Are More Beneficial than You Think



Shutterstock/retro67

By ULRIKE PARSON

About Intelligent Information and Omnichannel Distribution

The ongoing digitization of business requires us to rethink how we deliver user information. Technical documentation with hundreds of pages within one document is no longer viable. Today, we must deliver small content modules that show only the content that users need at a particular moment in time and in a particular context. Users need intelligent content or intelligent information instead of documentation.

Intelligent information dynamically adjusts to the context of the user and the use. It provides target-oriented information for all product lifecycle phases and various search and filter options. Intelligent information is modular, structured, and semantically enriched, and applications can evaluate it for specific usage scenarios.

From Multichannel to Omnichannel Distribution

Intelligent information is published and brought to users on various channels: as PDF, on the Internet, in social media, in mobile applications, or via chat bot or voice assistants. Based on the idea of single-source publishing, one piece of content is transformed into different output formats and can be presented in different media.

In today's interconnected world, the content that describes products is no longer split into silos like printed marketing materials, data sheets, websites, mobile apps, and technical documentation. The borders between the content types begin to blur and disappear as content is mixed on online platforms. Our users, of course, have never cared where a specific piece of information came from, as long as it has helped them with their question.

Today, users access product information on multiple channels, sometimes even simultaneously. They expect easy, invisible transitions between content types. For example, users want to be able to retrieve more detailed information about an advertised feature (marketing material) from the technical documentation (see Figure 1).

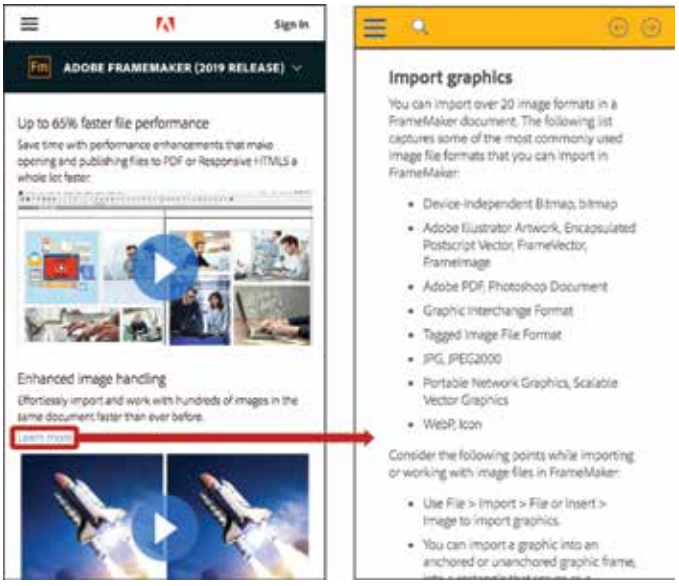


Figure 1. Example of linking marketing content with technical documentation content (Adobe).

Users expect the information in the different channels to be consistent regarding technical properties and terminology. Tone of voice and level of detail, however, might vary depending on channel and user. Content must be tailored to fit the user journey. A typical user journey could be to evaluate and compare product features and assess the quality of the documentation before making the decision to buy a specific product. During the journey, content must be findable and discoverable.

The difference between multichannel and omnichannel distribution is that we are no longer talking about content from a single source. Today, content is authored, edited, and published by multiple groups using different IT systems—typically by marketing, technical communication, product management, technical support, service, training, and sales. Each content group creates a different type of content. Keeping these different content types consistent and providing a smooth user journey is one of the biggest challenges today in content engineering. Another challenge is harmonizing technical formats and interfaces that are used to transport content between IT systems without creating a copy-and-paste hell for the authors.

How to Make Content Suitable for Omnichannel Distribution

So, how do we create content that is suitable for omnichannel distribution?

In short, we need to create content modules that are self-contained, usable, and reusable without any context. At the same time, the modules need to include all the information required to create context and to determine the purpose of the content. It's like creating a jigsaw puzzle with pieces that have content on the front, hidden instructions for use on the back, and little tabs for connecting to other pieces.

Properties of Omnichannel Content

Omnichannel content must have the following properties:

- **Granular**, meaning fragments and topics instead of documents
- **Reusable** for different audiences, contexts, and purposes
- **Enriched** with metadata for creating context and determining the purpose
- **Portable**, technically suitable for different media
- **High-quality**, fulfilling the information needs of the users—correct, understandable, appropriate, and accurate
- **Distinct**, separation between content and structure definition, content and layout, and content and metadata

The Importance of the Content Model

For an assembled jigsaw puzzle to look nice, all of the pieces need to follow the same design patterns. Otherwise the assembled image would look scrappy. The design of content is shaped by its terminology and structure. To reuse content across channels, we need to create content with harmonized content types that are tailored to specific information needs: instructions, procedures, feature descriptions, technical data, how-to videos, training units, exercises, and others. A content model with defined content types makes it easier for the authors to generate consistent

content and, at the same time, makes it easier for the users to recognize the purpose of the content.

Within the content, consistent terminology helps readers to understand and grasp the information quickly. And although terminology management has a long tradition in technical communication, a lot of organizations still have a long way to go to reach company-wide terminology management. How often have you seen a marketing brochure that speaks the same language as the user documentation?

The Importance of Harmonized Metadata

Let's go back to our jigsaw puzzle image. To assemble the pieces, the tabs at the edge of the pieces need to fit the corresponding blanks of other pieces. Also, we need the information on the back side of the pieces—who needs this information and when? Tabs, blanks, and back-side notes are represented by metadata assigned to content modules, as shown in Figure 2. Metadata supports content selection based on queries and assembly of information into larger documents.

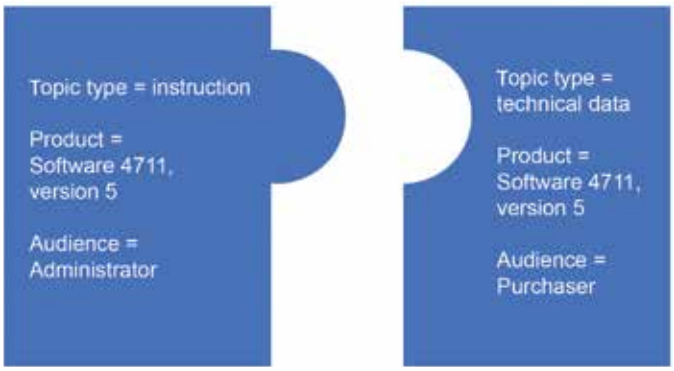

Topic type = instruction
Product = Software 4711, version 5
Audience = Administrator
Topic type = technical data
Product = Software 4711, version 5
Audience = Purchaser

Figure 2. Connecting metadata for content modules.

Just as terminology controls the language within the content, a harmonized vocabulary controls the language within the metadata. A corporate taxonomy is as important as terminology and content types. Otherwise, the search or filtering functions of the content delivery systems would have to consider all possible terms within metadata and relations between content modules. This would lower the quality of search results and make the assembled content less readable. By harmonizing metadata, we connect related content across discovery, access, and management.

How Do Standards Help Us?

Harmonizing content types and metadata is difficult enough when you have to do it in your own organization. To achieve this kind of harmonization across organizations, you need strong nerves. As digitization progresses, content needs to be orchestrated not only within one organization but also across companies and their products. Consider a smart factory where machinery and devices from different manufactures are combined in common processes. As a result, the information about how to operate and maintain this "machine park" needs to be assembled and aggregated from different sources, as shown in Figure 3. Smart factories require intelligent information to be delivered to the user in a common application, regardless of the source and original authoring format. This requires standardized delivery formats and taxonomies.

Figure 3. Intelligent information for smart factories.

This is where standards come into play. Standards help us overcome the obstacles that remain after harmonization by industry standards has reduced inconsistent content models and a variety of formats and interfaces.

Standards can refer both to technical solutions and methodologies. That means you can use a specific standard format or interface or a standard approach for developing your content model. Some standards are maintained by official standards organizations such as DIN, ISO, IEEE, and OASIS. Others are de-facto standards originating from individual companies.

Benefits and Disadvantages of Standards

Standards offer various benefits:

- Standards are usually developed by subject-matter experts from leading industry companies and reflect a lot of know-how and practical experience. Most standards are updated on a regular basis to reflect the developments within the industry.
- Standards come with documentation.
- Companies use standards as reference to harmonize their purchasing processes.
- Standards define specific quality criteria so that compliance implies good quality.
- Especially standards from international standardization organizations, such as ISO, are agreed upon within the industry.

There is a dark side, of course, because one size does not fit all. A standard must be generic to a certain degree, otherwise it won't fit everyone. A standard represents

a compromise between the individual interests of the contributors and stakeholders.

A good example of a compromise is the DITA standard. It contains different domains for specific purposes, such as the programming or learning domains. These domains reflect the interests of DITA users that need to write developer guides or training courses. Over time, attempting to meet the needs of everyone expands the core elements to such an extreme that the standard may become bloated and unwieldy for many potential uses.

Working in a standardization committee and developing a standard for any purpose can be a tedious task, because compromising takes more time and effort than just following your company's roadmap. This is the reason why industry standards tend to lag behind recent industry developments.

Existing Standards for Technical Communication Content

With regard to technical communication content, there are already a number of standards that we can use (see Table 1). Some of them are mere technical standards defining formats or interfaces, while others also define semantics and content models.

Consider the requirements in "How to Make Content Suitable for Omnichannel Distribution": from the list in Table 1, the DITA and iiRDS standards are most relevant for technical communicators:

- **DITA** provides both a file format and a semantic content type definition that are well suited for authoring content modules. Both make sure that all the jigsaw puzzle pieces follow the same design patterns.
- **iiRDS** provides standardized metadata that can be used to enrich content modules with context and purpose information. This information shapes the tabs, blanks, and back-side notes of our jigsaw puzzle pieces. In addition, the iiRDS package format allows content creators to deliver their content in a standardized way.

Both DITA and iiRDS provide machine-readable formats that can be processed and converted to fulfill the special publication needs of each channel and to feed search and filtering mechanisms. Both standards offer default specialization and customization mechanisms that enable companies to extend and adapt structures and vocabulary to their needs.

Challenges for Technical Communicators

Creating omnichannel content is a challenge that many companies face and most of them still have a long way ahead of them. Why?

- Their content is still stuck in silos and proprietary formats.
- Their systems do not exchange information or reuse content among each other.
- Their processes are not laid out for content collaboration.

On the other hand, our industry is beginning to change. We can reduce effort and investments by using established standards as a basis for developing a harmonized content model for the organization. The particular needs of the company and the users that are not covered by the standard models can be implemented and specialized on top of the industry standards—

Table 1. Standards for technical communication content

Standard	Purpose	Technical format	Content model	Body	Comment
DITA	XML data model for authoring and publishing user information	Yes	Yes	OASIS *https://www.oasis-open.org/standards#ditav1.3*	Widely used in the U.S., in Europe only in selected industries, e.g. semiconductors and software.
iiRDS	Ontology for user information and package format for content delivery	Yes	Yes	iiRDS Consortium *https://iirds.org/*	The iiRDS (Intelligent Information Request and Delivery Standard) ontology is defined in RDF, a format of the semantic web.
RDF	Technical standard for formulating logical statements about web resources. Standard for describing metadata	Yes	No	W3C *https://www.w3.org/2001/sw/wiki/RDF*	Part of Semantic Web family of standards.
SKOS	Common data model for sharing and linking knowledge organization systems via the Web. Designed for the representation of controlled vocabulary.	Yes	Yes	W3C *https://www.w3.org/2004/02/skos/*	Part of Semantic Web family of standards.
schema.org	Collection of open standard vocabularies for different domains	No	Yes	Community activity *https://schema.org/*	

see Figure 4. This also makes proprietary solutions leaner and easier to exchange.

Figure 4. Using industry standards as basis.

Conclusion

To produce and distribute omnichannel content, organizations must adapt their content workflows, bridge their IT systems, and harmonize their content models.

There are crucial benefits when we use industry standards as the basis for these new workflows.

- We leverage the knowledge of the experts who contributed to the industry standards.
- We make workflows and IT systems less dependent on specific vendors and systems, as well as more interchangeable.
- We enable content synchronization across organizations and products.

For technical communication content, there are a number of standards. DITA and iiRDS, in particular, are a good match, because together they cover content and metadata models.

Organizations should invest in the development and maintenance of industry standards as part of their research and development activities to facilitate modern standards that provide a common ground for innovative content solutions.

ULRIKE PARSON (ulrike.parson@parson-europe.com) *is the CEO of parson AG, located in Hamburg and Berlin, Germany. She looks back on more than 20 years of professional experience in technical communication. She is specialized in content strategy, developer documentation, and Agile project management. The clients of parson include international and regional companies from various branches, such as software development, semiconductors, logistics, education, and healthcare.*

Ulrike is Steering Committee chair for the iiRDS Consortium that develops an international standard for delivering intelligent content.

CERTIFIED PROFESSIONAL Technical Communicator

CPTC™ PRACTITIONER

PRACTITIONER CERTIFICATION

OPEN NOW

Certified Professional Technical Communicator™ (CPTC)

Be a leader. ***Have your Foundation Certification? Take your career to the next level by obtaining your Practitioner Certification.***

For more information about certification and to start the process, visit *www.stc.org* or email *stc@stc.org.*

Learn more at www.stc.org/certification

Shutterstock/Visual Generation

Autonomous Agile Teams and Enterprise Content Strategy: An Impossible Combo?

By TOM JOHNSON | *STC Senior Member*

AGILE SOFTWARE DEVELOPMENT seems to favor independent, autonomous teams. In contrast, enterprise content strategy and engineering practices look to harmonize content across multiple teams and boundaries. In a software development model where Agile dominates as the norm, how do you reconcile the larger need for enterprise content strategy?

Agile Versus the Enterprise

As our documentation teams at work have grown and fragmented, split and combined, often driven by the needs of Agile teams spun up around new projects, it's caused me to think about how Agile fits in with enterprise content strategies. Agile teams tend to be independent, autonomous, smallish groups. They are single-threaded so they can be nimble and execute on a solution from end to end.

For example, companies might spin off a group of 20 people in various roles—engineers, testers, UX, project managers, and sometimes a technical writer—to build out a new solution for a long-shot project. Agile is a driving force that seems, at least in larger companies, to create startups within the enterprise. You need to move fast to stay competitive in a rapidly evolving tech landscape. Enterprises don't move fast; startups do. You give the startup complete autonomy to deliver a solution, without any roadblocks.

Agile software engineering and product teams sensing extensive documentation needs tend to hire a dedicated technical writer specific to their team. These writers often arrive with only marginal obligations to adhere to wider enterprise concerns. In large enterprises, tech doc teams might be walled off from each other entirely by business lines, unaware of each other's existence. Your Agile team (startup) might be the extent of your "company."

If all writers have to plug into the same toolchain, taxonomy, style guide, reusable content source, approval process, and publishing workflows as every other writer in the enterprise, then you've slowed down the tech doc process and are at odds with the nimble and quick pace of Agile development. Given that most modern software development follows Scrum (a subset of Agile), tech writers plugging into this same methodology usually have to sacrifice ideals about enterprise content harmony to thrive in their Agile teams.

Enterprise Content Strategy and Engineering

In contrast to the independent, autonomous efforts of Agile teams, enterprise content strategy has much broader, more encompassing goals. Enterprise content strategists and content engineering teams might ask how the documentation produced by a single team fits within the larger enterprise context. This strategist would consider all the content touchpoints customers have across the enterprise through their product journey, from pre-sales touchpoints with marketing to touchpoints with solutions engineers and business development to post-sales touchpoints with technical documentation, support, troubleshooting, and more. What is the customer experience from end to end? How do you ensure consistency of terminology, style, and message across all of these enterprise lines?

Search is where enterprise content intersects—where customers see all content mixed together, often juxtaposed in jarring ways. Enterprise content strategy, by definition, forces you to cross department and team boundaries to coordinate on a higher level. For example, content engineers might strive for a unified taxonomy for all enterprise content so that it can feed into a larger faceted search or content management system where assets are described, shared, tracked, and re-used. Faceted search simply doesn't work if all teams aren't on board with the same terminology.

Given the compelling needs for coordination at an enterprise level, is Agile at odds with enterprise content strategy? If so, are the benefits of enterprise content strategy strong enough to overpower the driving forces of Agile? They seem incompatible. Let's examine the cases for independent documentation and for unified content practices.

The Case for Independent Documentation

First, let's consider the case for independent documentation, such as that produced by an Agile team that isn't integrated and aligned within any larger, enterprise model. Any writer has to draw lines around their stewardship to survive, right? Take on too much, drawing the circles of ownership too large, and the work will crush you. You aren't paid, most likely, to produce anything beyond the needs of your immediate business unit. The business lines of the organization define who you're responsible to and what you're responsible to produce.

Sure, you could scour all enterprise documentation and champion a case across all doc groups (perhaps across hundreds of technical writers in your organization, in many different business units, all reporting along separate business lines) to fall behind a common style guide, taxonomy, tool chain, and approach in documentation, but why? You're paid to write documentation for a product from a particular business unit. Everything else is extracurricular. The additional effort to unify content across the enterprise will exhaust bandwidth you don't have.

The Case for Unified Content Practices

Focusing your vision only on your Agile team's needs, on the other hand, will likely lead to poor quality content. Content produced separately by many different teams, written in different styles, published by different tools and systems, without any larger awareness often results in a fragmented and disjointed content landscape.

Setting aside documentation concerns, engineers who build in isolation face similar problems. If engineers build separate systems that don't integrate, the user experience also ends up being disjointed and frustrating. Users might find that one SDK doesn't integrate with another, or that one tool is built on a technology that is incompatible with another.

At a previous company where I worked, customers were fed up with disparate systems that didn't integrate with each other. When two major customers (who brought in millions in revenue) threatened not to renew, it caused shockwaves across the company. Business leaders halted the development of all new products as software teams undertook efforts to make the existing software work as a harmonizing suite of applications.

A hodgepodge of technology is understandable in mergers and acquisitions, but few users will be patient with the idea that the single company they interact with is actually made up of dozens of small, independent, internal "companies" and that each of these companies doesn't seem to know what the others are working on or building, because none of the products work together.

As a worse case, in massive companies, totally isolated teams might even be working on different solutions for the same problem, unaware of each other's existence. Good documentation can't be written with blinders on.

Harmony Is Hard

I wrote about the need to make information harmonize in the larger landscape in "Principle 3: Ensure information harmony in the larger landscape" (Johnson 2017) from my series on Simplifying Complexity. The basic principle is this:

> *Before adding new topics to an information landscape, look to see how the new information will fit in with the existing information—across all information forms, from docs to blogs, forums, white papers, and more. Synthesizing information to harmonize with the larger information landscape requires wide reading and analysis but is essential for the user experience, since users often bounce from one information source to another as they consume information.*

Without question, it's much harder to understand the larger landscape in which you're writing and to assess how the content (the terms, styles, semantics, etc.) you're using fits into other content which you likely do not own or over which you do not have stewardship. It's easier to write a standalone article and push it out, like a blog post.

Strategies for Unifying Content

One strategy for unifying content involves bringing all tech writers into the same enterprise component content management system (CCMS) so they can see where content overlaps, re-use existing content in granular ways, find ways to leverage a common taxonomy, and more.

Although I often hear about the benefits of enterprise content in a CCMS, I've never seen it actually implemented in large companies. Centralizing content from the top down requires an engineering group to provide enterprise-wide tooling, yet each business unit is usually responsible for its own needs. One group's needs don't always align with another's.

Even if some central governing body were to require all tech writers to use the same authoring and publishing tools, such an approach might stifle innovation. Writers would likely be stuck in the same systems for their entire corporate lives, with little ability to grow and evolve tools. Why not let these startups compete naturally with each other and see which one wins, drawing others to its side naturally rather than in a compulsory way?

Another strategy for allowing autonomous teams to harmonize within the larger content landscape is to create common glossaries and taxonomies. These can be created outside of any common tools. Much of the clarity and meaning in technical documentation is tied up with the terminology writers use. Control the use of terms at the source and you have a shot at making content (even if in different tools and systems) flow together more harmoniously.

This is easier than it sounds—content by nature is messy. Content on any documentation site might all read with the same general tone and follow similar structures (due to convention with industry standards), but in many places the content might be muddled and confusing. The content might use different terms to refer to the same process or have functionality that overlaps, or it might present solutions without adding the context of similar or contrasting solutions within the same space.

For example, during a recent project, I realized that the term "video skill" was used in different ways in the community and across other doc groups. My awareness of the term's problematic meaning only surfaced as I started reading beyond my immediate group and user community. There wasn't an easy solution to the problem, but at least I acknowledged this point of confusion in the documentation.

Independent documentation that does not consider anything beyond itself might be acceptable, just as a C or B- might be a passing grade on a test. If you want an "A" on your documentation, increasing its quality, you have to increase your sense of awareness (see Figure 1).

The first level of awareness is to master your own product. Understanding your product as a subject matter expert (SME) is practically required to write detailed documentation in the first place.

The next level is to be aware of documentation that other technical writers in other groups are working on. This is more applicable in larger companies with more than just a few tech writers. Good writers expand their horizons to understand docs they don't necessarily own but that have some influence or relationship with their docs.

The next level is to be cognizant of content outside the documentation domain—content on blogs, white papers, ebooks, business development slide decks, and other collateral produced by groups outside of documentation. Groups like field engineering, business development, support, and so on are, in fact, producing content. Technical writers often aren't aware of this content.

Stretching awareness even further, a good writer should also be aware of the competitor's landscape. I wrote about this in "My Documentation Takeaways from the Boeing Disaster—Two Essential Doc Questions to Ask for Any Project" (Johnson 2019). In that post, I argued that good documentation should address the question, "How does this product differ from other products?" This question is rarely addressed in documentation since many companies do not call out competitors or compare their products with competitors.

This is one area where practitioners can learn from academic genre conventions. As a required section in academic articles, before diving into the details of one's argument, nearly every academic article presents the context of previous research that has been done on the

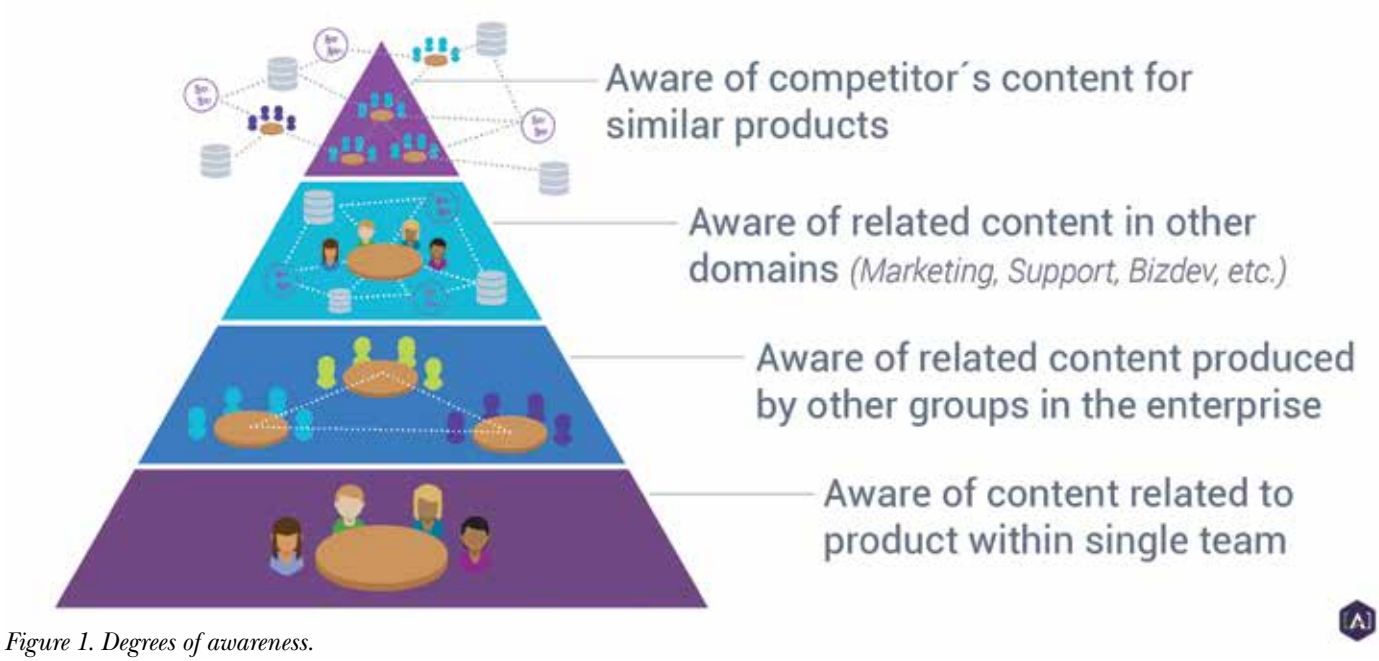

Aware of competitor´s content for similar products
Aware of related content in other domains (Marketing, Support, Bizdev, etc.)
Aware of related content produced by other groups in the enterprise
Aware of content related to product within single team

Figure 1. Degrees of awareness.

topic. Academics don't just start writing without awareness of what has already been written on the topic. They start by surveying the landscape, summarizing what other research has been done, and then moving forward with their current argument. In documentation, this kind of landscape survey might involve mentioning how the product fits in the enterprise and competitive landscape of similar technologies.

Conclusion

Most writers won't expand their horizons to consume more than their immediate product documentation requires. No one will force you to read documentation written by other departments or docs written by competitors. So should you even bother? Surely this expanded analysis isn't scoped into your project plan or expectations, right? Where do you find the time or bandwidth to do it?

I'm not sure. Those who write our paychecks often won't factor in time for this, especially given the time you already need to ramp up on the product's technology itself (which is not insignificant in the developer documentation space).

Although I want to champion enterprise content strategy, if the Web has shown us the future, the future is probably independent publishing from disparate groups. It works out all right for the most part on the Internet, though enterprises might have another standard. For example, you won't see articles in WebMD repurposing content from Yahoo Health covering the same topic. When you search for content, you get a lot of different hits on the same topic, with different approaches and styles and perspectives from different authors at different times on different websites. Content on the internet is redundant, outdated, disjointed, written in different styles for different purposes, audiences, business contexts, and more. But it works out all right—users find what they need, ignoring content that doesn't quite match their search.

In the tension between Agile and the enterprise, most likely one embraces a middle ground here—finding some awareness of other content but not extensive awareness, reusing content within smaller groups in the same business units rather than across the enterprise. In general, good writers will read beyond their Agile team's boundaries to provide better semantic context and fit with their content, but none of us has time to read endlessly and widely before the doc is due and another project takes the stage.

Whatever your position, if you want to write more informed documentation, try to climb up the pyramid of awareness. Start with awareness of content related to your product, and then as time permits, expand to learn about similar content across the company, then look at the same content in other business groups, and finally expand your analysis of competitor documentation for similar products. Most technical writers complete the base layer and maybe the second, but few climb up higher.

TOM JOHNSON *(*tomjoht@gmail.com*) is an STC member living in Sunnyvale, California. He works on developer documentation at Amazon and writes a blog on* idratherbewriting.com. *He also has an API documentation course available at* idrather bewriting.com/learnapidoc.

REFERENCES

Johnson, Tom. "Principle 3: Ensure Information Harmony in the Larger Landscape." *I'd Rather Be Writing.* 2017. *https://idrather bewriting.com/simplifying-complexity/ensuring-information-harmony-in-the-larger-documentation-landscape.html.*

Johnson, Tom. "My Documentation Takeaways from the Boeing Disaster—Two Essential Doc Questions to Ask for Any Project." *I'd Rather Be Writing.* 20 March 2019. *https://idratherbewriting.com/2019/03/20/two-doc-takeaways-from-boeing-disaster/.*

Product Answers: Engineering Content for Freshness

By MEGAN GILHOOLY

Shutterstock/Viktoria Kurpas

CONTENT ENGINEERS FORM the structure for content, helping to shape systems and processes that allow content to be created, transformed, translated, and published to many interactions—*and* providing new capabilities to keep content "fresh." In this article, we'll explore a definition and approach to "content freshness" made possible through content engineering.

An online search for the term "content freshness" yields an endless list of search results covering how fresh content can dramatically improve SEO. Of course, maintaining great SEO is important, but the need for fresh content goes beyond search results. Unless you're on the hook for lead generation, you should be far more concerned with the other benefits of freshness to maximize the value you are squeezing out of your content.

Achieving fresh content requires more than links on your website to newly created content. In fact, maintaining freshness can be quite complex. Luckily, a well-thought content architecture, with modular topics and appropriate reuse, can help.

What Makes Content Fresh?

Content freshness means many things, and how you define it will depend on the type of content you produce, the underlying objectives of the content, how it is distributed, and the overarching content strategy. In general, though, *fresh content is up to date, timely, and pertinent, and provides new or different insights to existing information*. Let's break that down.

Up to Date

Fresh content might be recently updated, but there's a difference between recently updated and up to date. A topic describing a policy, for example, is unlikely to change much over time. Of course, it's important to review regularly to ensure it remains up to date, but unless you have good reason to change how the policy is written, you won't update the topic itself. On the other hand, content that describes frequently changing features of a product probably does require regular updates, for example, possibly every 3-6 months. So while SEO enthusiasts will debate whether or not recently updated content ranks

higher, your readers only care whether technical product information is current.

Timely

Fresh content is relevant to today's trends. For example, consider the very term we are discussing here. If you search for "what is fresh content" on Google, you'll get a long list of prescriptions for building fresh content for SEO. SEO is important, but for customer-obsessed content producers, it's not the end game. Unless you're specifically wondering how to use fresh content to boost your Google rankings, the majority of the search results will not be helpful. Because fresh content for SEO is all the rage, however, the high rankers are timely.

Pertinent

Fresh content provides desired insights and surfaces them at the appropriate time in the customer journey. In other words, customers can find information that is relevant to their current needs. For example, a security engineer in the consideration phase of the journey needs detailed specifications to determine whether or not the product meets security standards. A developer trying to integrate your product into their current environment for the first time needs APIs and an implementation guide. And a support agent trying to fix issues after an upgrade needs troubleshooting content. Fresh content is engineered to be accessible to specific customers at specific touchpoints.

New or Different

Fresh content provides previously unknown information or adds a new perspective to a topic. For example, fresh content might describe a new feature, or it might explain a new use for an old feature. It might provide a new solution for an existing problem, or it might suggest new insights learned from using a previously published solution. And content that designates slightly different steps for a Web app that just launched as a mobile app would certainly be considered fresh.

How Does Structured Content Help?

Structured content encourages content teams to:

- **Reduce** the size of content produced and get smaller chunks of content flowing more quickly through the production cycle;
- **Reuse** topics across multiple content assets to optimize the value of each topic and ensure consistency of messages across channels; and
- **Refactor** content by altering its internal structure in ways that are imperceptible to customers.

Defined by an information model (IM), a unified content structure separates the management of content (getting the words right) from its ultimate delivery format (the way the words lay out on a page or screen). Structural standards emphasize creating content in easily digestible chunks to form content patterns that are so predictable in size, organization, and construction that any publishing script can reference the words from a single source and publish them in any format.

With this foundation in place, you can write once, edit once, translate once, and publish to multiple places with the push of a button. With the right mechanisms, you can also ingest content from multiple locations, transform it to a standard structure, and manage it from a single source.

Maintaining Freshness

Structured content makes it easier to maintain freshness of content in many ways.

Single Source

Structured content supports reusing topics from a single source, making it easy to update the topic in one place and publish to many places. By single-sourcing your content, you mitigate the inconsistency of having new information published in one place while contradictory, old information is published in another. Managing small, modular topics and reusing those topics from one source simplifies the work of ensuring your content stays up to date.

Deliver Pertinent Content

Effective structured content includes well-planned and intentional tags within the topics themselves. You can use these tags to personalize the content experience, delivering pertinent content to your customers based on who they are, what they do, and where they are in the customer journey.

Dynamically Link Relevant Topics

Using the hierarchy of structured content, you can automate a "related topics" section of your delivery platform that will dynamically update as your team updates or adds content to the portal. Of course, you can do this manually without structured content, but the effort involved in maintaining manual lists of related links can quickly overwhelm a team.

Mitigate Inconsistent Updates

A successful structured architecture, managed by the right component content management system (CCMS), enables content producers to easily find related content that might also require updates to maintain consistency across a content ecosystem. A primary role of a CCMS is to indicate all the published content that will be impacted when editing a topic. In this way, structured authoring supports productivity by eliminating the need for authors to do exhaustive manual searches for related content changes. The net result is fewer inconsistencies and more up-to-date content.

Impact of Content Freshness

While it's important to think about how structured content makes it easier to keep content fresh, ultimately, the power of fresh content is in the customer experience.

Customers use product content throughout every stage of the customer journey (see Figure 1), making it a crucial element in the customer experience. Customers want product answers, where and when they need them. Making customers wade through stale content has some pretty dire consequences:

- Customers can't find the answers they're looking for and proceed to use your product incorrectly or miss using it to its fullest potential
- Customers spend an excessive amount of time sifting through different versions of content before finally finding relevant and accurate answers
- Customers give up on self-service and call support to get an answer

Benefits of Fresh Content

Stale content can damage your brand and perception of your products, lower customer satisfaction, and create churn. Fresh content increases customer trust, engagement, and loyalty, while reducing unnecessary scrutiny and risk.

Increase Trust, Engagement, and Loyalty

Customers know when they find good content. And when they consistently find the content they want or need on your site with minimal effort required, they learn to trust your brand. When they trust your brand, they will continue to engage with you, driving loyalty over time. In this way, fresh content helps build customer trust, engagement, and loyalty, which can be a significant competitive differentiator.

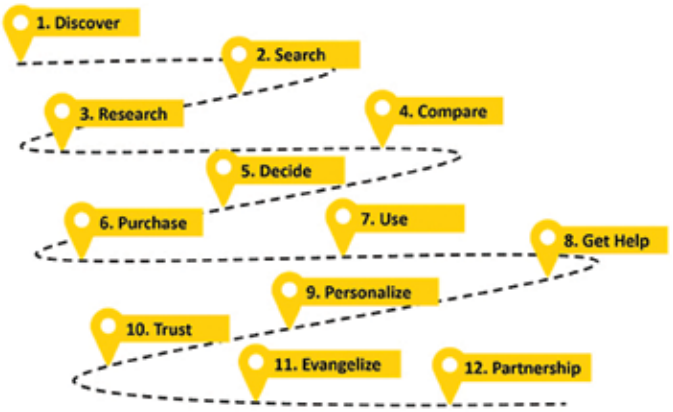
Reduce Scrutiny and Risk

Stale content leaves a company open to scrutiny by competitors, analysts, and customers who might highlight a lack of features or functionality by relying on outdated documentation. This can have a significant but silent impact on revenue for a company. For example, potential customers use analyst reports to vet your product. If outdated information leads a trusted analyst report to underrepresent your product, potential customers will see (and believe) that outdated information for an entire year. You can avoid this revenue-impacting situation by ensuring that your content remains fresh.

Delivering Fresh Content

Now you know you need to deliver fresh content and you know that structured content will help, but delivering fresh content goes beyond managing your content. Your content and your delivery portal should work together to deliver the following features and characteristics.

- **Dynamic delivery**. Your content portal should deliver different content to different users at different times, and what it delivers should change automatically based on tagging as your content changes.
- **Analytics**. To truly understand how fresh or stale your content is, you need an analytics tool that includes content utilization (how much topics are used), content aging (how long has content been published without updates), and user retention rate (percentage of users who continue to use a topic for a specified number of days after an event).
- **Subscription**. When users subscribe to content, there are two freshness benefits. First, users opt into content that is pertinent to them, so you never have to wonder. Second, content with high subscription rates deserves more attention and effort in maintaining freshness.
- **Automatic expiry date**. For content that is highly sensitive or frequently updated, configure your content portal to automatically unpublish content that has reached a pre-set expiry date (for example, six months or one year). The portal must also automatically log a task for your team to check this now-unpublished content to determine if it should be republished as is, updated and republished, or deprecated.
- **Immediate feedback from users**. Give users a form—integrated with your ticketing system—to submit feedback to you and generate automated tickets to solve issues. By empowering customers to tell you when content is stale, your team can manage content you might have missed.



Zoomin Software 2019

Figure 1. Customer journey stages.

Content freshness goes far beyond updating content to drive SEO. Your customers rely on your content and expect it to be fresh. A well-thought-out content freshness strategy includes the two-pronged approach described in this article: engineering structure into your content to facilitate appropriate content architecture and tagging; and developing a digital experience that delivers up-to-date, timely, and pertinent information that customers value.

MEGAN GILHOOLY (megan.gilhooly@zoominsoftware.com) *recently joined Zoomin after serving as Senior Manager of Content Management and Strategy at Amazon and AWS. Previously, she was Director of Information Experience at Ping Identity, and Director of Technical Communications at INVIDI Technologies. Connect with Megan on Twitter* (@megangilhooly) *and LinkedIn* (linkedin.com/in/megangilhooly).

Registration for the 67th Annual Technical Communication Summit Is Now Open

"IF YOU ARE PAYING ATTENTION to the massive global shifts affecting us as organizations and individuals, not only do we need to meet, we need to attend meetings that provoke thought, define forward-thinking, actionable learning, and provide a toolkit for participants to move ahead in ways which will positively affect business outcomes." Tahira Endean, CMP, DES, CED, from Intentional Event Design.

Join STC and fellow colleagues at the 2020 Summit, the premier conference for technical communication education and networking, 15-18 May at the Hyatt Regency Bellevue, WA!

Registration for this can't-miss event is now open, and STC is offering a special registration rate of **only $975 for STC members, an $800+ savings** from the regular nonmember rate! Your conference registration includes the Welcome Reception on Sunday, continental breakfast on Monday and Tuesday, the Closing Honors event on Wednesday, refreshment breaks, access to unparalleled education sessions, and information on the latest industry tools and technologies.

Shutterstock/Cascade Creatives

STC is also pleased to announce the Cornerstone Opening Session Keynote is Dona Sarkar and the Capstone Closing Session Keynote in Marie Zimenoff. Visit *summit.stc.org* to learn more about them.

More exciting news and speakers will be announced in the coming months. So what are you waiting for? Register today! See you in Bellevue!

Subscribe to STC Roundtable

DID YOU KNOW that when you subscribe to Roundtable at the monthly or yearly rate, you'll get access to the complete backlog of content? That's 8 months of content you'll be able to access immediately.

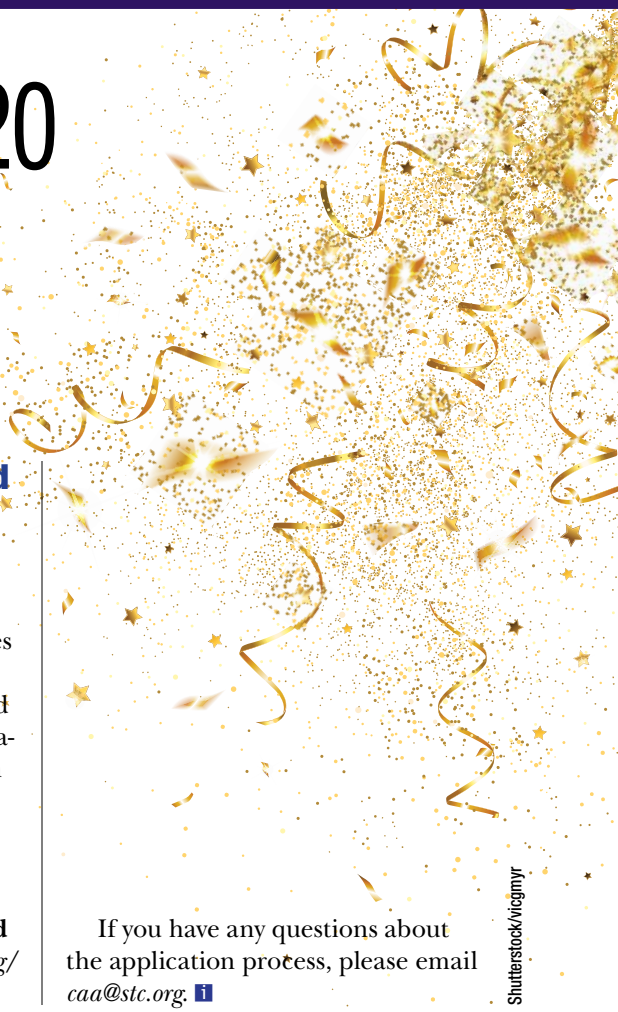
Not yet convinced? Here's a list of the content that's been featured in Roundtable since its launch in March.

- **March:** The Future of Technical Communication, curated by Scott Abel, "The Content Wrangler," an internationally recognized content strategist.
- **April:** Content Strategy and Experience, curated by Rahel Bailie, an STC Fellow and Director of Content at Babylon Health.
- **May:** Professional Development, curated by Alyssa Fox and Alisa Bonsignore. Alyssa is a former STC President and a content strategist and marketing leader. Alisa is a writer and researcher who clarifies complex ideas from her office in the San Francisco Bay Area, and currently serves on the STC Board of Directors.
- **June:** Leadership and Teams, curated by Toni Mantych, Senior Director of Product Content Experience at ServiceNow and a recognized Top 200 Content Experience Strategist.
- **July:** Content Design and Development for Machine Delivery, curated by Toni Byrd Ressaire, a consultant, trainer, and information developer. She founded the company Info4Design and is Co-Founder of the Information 4.0 Consortium.
- **August:** Content Measurement + Intelligence, curated by Colleen Jones, author of *The Content Advantage* and Founder of Content Science, a content strategy and intelligence firm.
- **September:** Globalization/Localization, curated by Karen Tkaczyk, an award-winning freelance translator, editor, and trainer.
- **October:** Learning Experience Design, curated by Phylise Banner, a learning experience design pioneer with 25 years of experience.

Plus, you'll get access to all future content! Here's what's in store for the rest of 2019.

- **November:** Content Engineering, curated by Cruce Saunders, a content experience strategist and Founder of [A], a Content Intelligence firm.
- **December:** Information 4.0, curated by Ray Gallon, an STC Associate Fellow, and President and Co-Founder of the Transformation Society.

Nominations Open for 2020 Community Achievement and Pacesetter Awards



Shutterstock/vicgmyr

Community Achievement Award

The Community Achievement Award (CAA) recognizes a SIG's or a professional or student chapter's outstanding accomplishments in achieving the Society's goals through a wide range of programs and activities. All communities are encouraged to apply. The CAA process determines the prestigious Community of the Year award.

- **Due date for 2019 nominations:** 27 January 2020
- **Get the application and guidelines here:** *https://www.stc.org/community-achievement-awards/*

Community Pacesetter Award

STC's Community Pacesetter Award recognizes innovative and successful community initiatives. Unlike STC's other Community Achievement awards, which recognize communities for consistent strength in many varied activities, the Pacesetter Award recognizes the successful implementation of a single, beneficial innovation that may be implemented by other STC communities.

- **Due date for applications:** 24 February 2020
- **Get the Web form application and guidelines here:** *https://www.stc.org/community-pacesetter-awards/*

If you have any questions about the application process, please email *caa@stc.org.*

Nominations Open for the 2020 Jay R. Gould Award

THE JAY R. GOULD AWARD for Excellence in Teaching Technical Communication honors the distinguished teaching career of professor Gould. Professor Gould's academic mentorship of technical communication students guided many men and women into the profession. The Jay R. Gould Award for excellence in teaching technical communication honors his distinguished teaching career, which is an outstanding example of the long-term commitment, innovation, and excellence in teaching. This award honors true academic mentorship; a record of successful students, defined as those who are involved in STC, and actively working in the profession; involvement in student activities outside of the classroom; innovation and creativity in teaching, and involvement in research that leads to changes in the way technical communication is taught.

- **Who is eligible:** STC Members who have belonged to the Society for eight years or more, been involved in post-secondary education for at least fifteen years, and whose primary career is teaching the philosophy and theory of technical communication as a distinct discipline.
- **Award Description:** *www.stc.org/jay-r-gould-award-excellence-teaching/*
- **Who may nominate:** department chairs, deans, colleagues, students, and other STC members acquainted with the nominee.
- **Nomination process:** please send your nominations to: Sally Henschel, Jay R. Gould Award Committee chair: *sally.henschel@msutexas.edu*
- **Due date for nominations:** 15 November 2019
- **Due date for applications by nominated candidates:** 17 December 2019
- **Honorees notified:** After vote by STC Board of Directors, February 2020
- **Award Presented:** At the Honors Event at the STC Technical Communication Summit
- **Previous Jay R. Gould Winners:** *www.stc.org/jay-r-gould-award-excellence-teaching/jay-r-gould-award-previous-winners/*
- **Questions:** Please contact Sally: *sally.henschel@msutexas.edu*

Defining Intelligent Customer Experience

Shutterstock/Rawpixel.com

BY CRUCE SAUNDERS

THE "NEXT GENERATION" of customer experiences is already here, but it is unevenly distributed in the way that the future often shows itself. Customer interactions get more intelligent, responsive, and personalized every month. The organizations leading the way are intentionally creating "intelligent customer experiences."

What is intelligent customer experience?

An intelligent customer experiences (ICX) is a uniquely assembled interaction between an organization and a customer in which the content responds to the customer's individual context. With ICX, organizations automatically form adaptive relationships by listening to customer data and responding with contextually relevant content in real time. These new-era experiences deliver meaningful and useful content across channels, reducing the manual effort and transaction friction for consumers.

Intelligent customer experiences have five characteristics, as shown in figure 1:

- Omnichannel: Assembling content modules into orchestrated, unique customer experiences across multiple channels, platforms, and devices
- Contextual: Listening to and understanding a customer's context and making that customer data available to real-time sessions. Bringing together relevant modular content from a content pool to meet customers in their required formats and views
- Fluid: Flowing across a complete content supply chain, customer experience (CX), and martech software stack, and into multiple interfaces and renderings with minimal rework
- Automated: Responding using conditional logic to facilitate instant and long-term customer journeys without human intervention or connecting with robot consumers of structured data
- Conversational: Connecting with customers through intent and response interactions—including chatbots and voice interfaces—facilitated by a combination of human and machine intelligence

Publishers face many challenges creating a new operating model and orchestration approach for dynamic omnichannel content. This column addresses the changing content ecosystem and the evolving dynamics in the space. Questions, suggestions, or feedback? Email *editorial@simplea.com*.

To get *there*, we need to engineer *here*

To achieve the next generation of customer experience, content must first be *available* for assembly.

Not every content interaction needs to be dynamically assembled, nor does every customer experience need to be "intelligent." It's okay for some content to sit in big chunky fields published on traditional pages. Books are wonderful and should always exist. Compiled digital renderings (such as PDFs) serve a useful purpose. Highly designed, image-laden, pixel-perfect web interfaces are not going away, but more and more hybrid views on our content can—and should—be assembled from components.

Intelligent content cannot reasonably be separated from intelligent customer experience. The essential ingredient to all customer experience is content, thus *content problems impact customer experience.* More and more, modern customer experiences rely on structured, intelligent content.

Producing and maintaining *intelligent* content is directly derived from the practice of content engineering. The foundation for ICX is people, process, and technology purposefully aligned and intentionally using standard content and semantic structures for translation between roles and systems.

Not only is the engineering of intelligent content the key to ICX, but it is also the only sustainable way to effectively, efficiently, and productively scale authoring and publishing operations.

Paying diligent, ongoing, scientific attention to the structure, formation, assembly, and delivery of intelligent content is the domain of the content engineer.

Personalized experiences are assembled, not made

In the traditional approach employed across many digital experience teams today, content is crafted as pages or interfaces and delivered to a single endpoint. In the new era of ICX, content must perform work across more than just one page, interface, or customer interaction. We need to craft systems of content portability that content leaders can leverage to activate many experiences from one content pool.

To assemble an intelligent customer experience, we bring together modules of structured content into different representations, as shown in figure 2, so that, for example, a product and related descriptions can assemble into many variant outputs. This idea might seem like common sense, but implementations are still too rare.

First, we need to get to a basic structure for reuse. Then, beyond simpler presentation variation comes the more advanced contextual personalization of content. Personalization depends on structured content meeting real-time customer context. There are, however, some common misconceptions about what makes personalization possible.

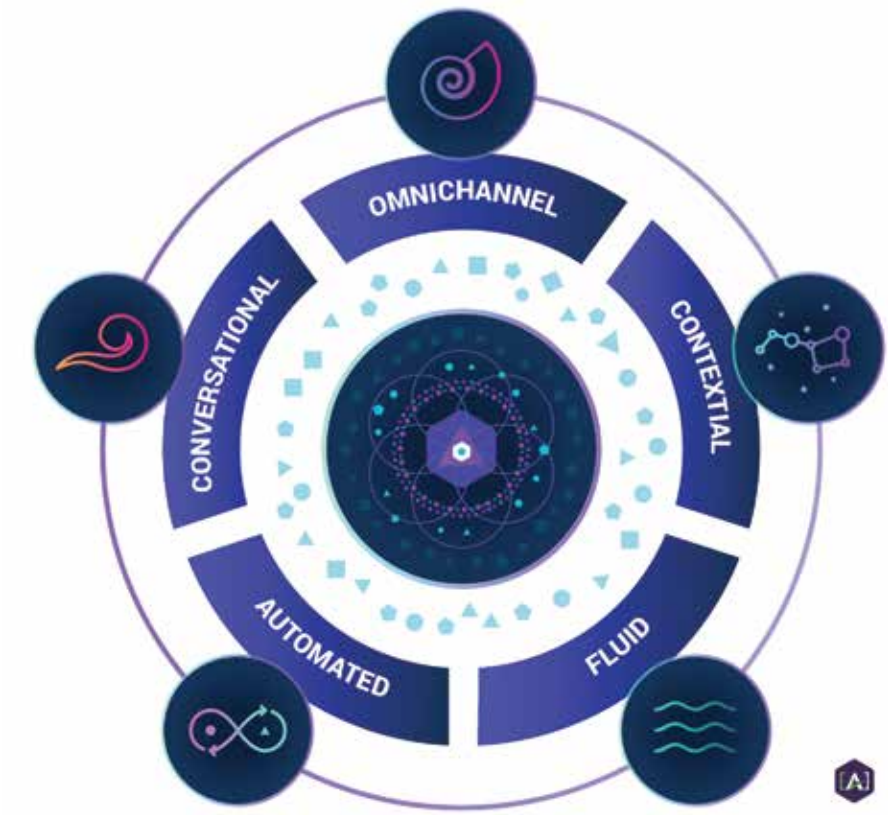


Figure 1. Intelligent customer experience characteristics

Buying technology

Personalization is not a piece of tech. No software gets us there. Those technical solutions boasting that they can "manage and deliver content across multiple channels," all depend on content structures and semantics, which all relies on content engineering. When we write content to a single platform, it's stuck. Content engineers work to decouple content from systems so that systems are less constraining and restrictive to the content supply chain. Content needs to move fluidly between systems and representational states.

Hiring a pro

Personalization is not accomplished by making a single hire. No personalization guru can do much real work without structured content. We can make really intense spreadsheets of personalization characteristics and data, but unless the content can keep up with the targeting profiles and algorithms, they are all paper tigers. Hiring is only part of the answer.

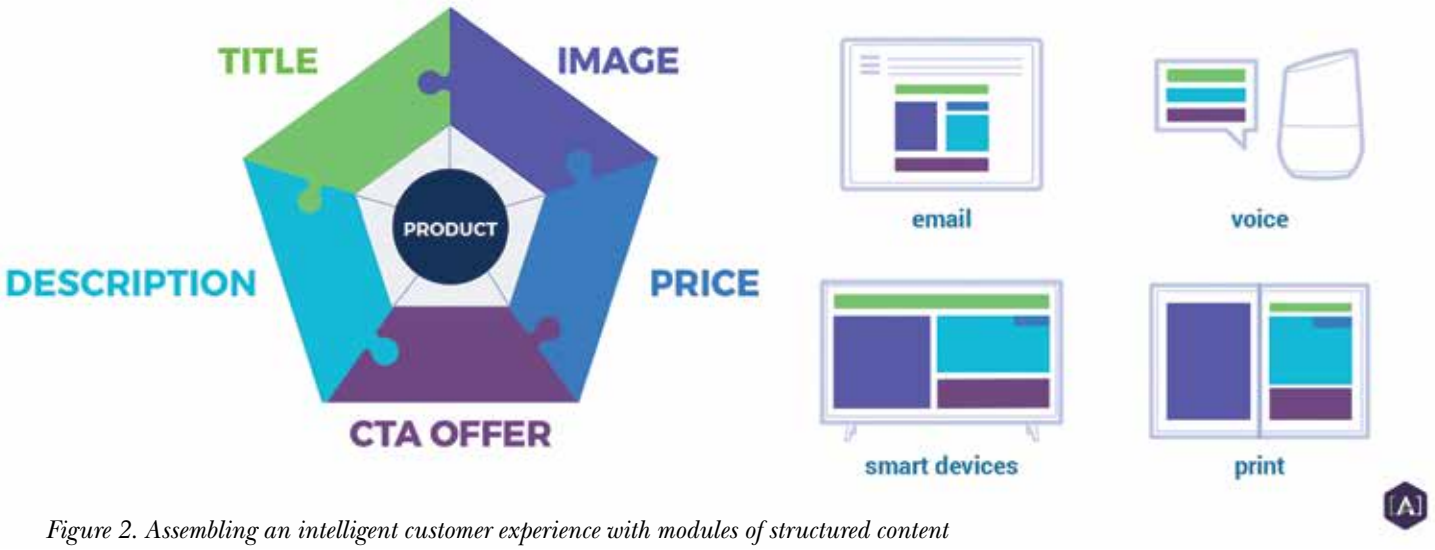


Figure 2. Assembling an intelligent customer experience with modules of structured content

Changing the process

Personalization is more than just creating a new process or changing an old one. It requires process, but process change alone misses the goal. Technology does enable reduced content supply chain friction, when that technology is architected to help, and engineering content and technology must be included in the overall program.

Personalization is an effect born from many causes across content and context, across strategy, engineering, and operations. We should not assume intelligent customer experiences will happen by pressing any one button, buying any single technology, or hiring any one person. Shortcuts lead to short impacts. This next generation of experience requires us to deliver integration across various native content sets at scale. To meet this challenge, content will be engineered and orchestrated on an entirely new level, and it requires us to work toward the whole, one part at a time. Getting there takes leadership.

Content leaders drive intelligent customer experience

Over the last two years, [A] has seen more and more professionals with a background in technical communication, content strategy, and the technical dimension of content becoming significant leaders in organizational change. We also seen content leaders taking up global portfolios as corporate Vice Presidents, like Anna Schlegel at NetApp. Introducing the practices of content strategy, content engineering, and content operations into teams takes work and perseverance, but it is happening.

The enterprise landscape keeps changing. Customer experience initiatives are taking over marketing and technical communication organizations. We see omnichannel customer experience moving from individuals' roles into organizational structures. The wheels are in motion. Content sets will eventually be connected across a single customer journey; it's just a matter of how and when.

Knowledgeable content professionals bring the wisdom and tactical knowledge necessary to move the conversation away from a one-dimensional discussion of tools toward a more systematic approach to content intelligence. The following are some things we can each do.

- Invest in or advocate for *skill development* around content engineering to grow the practice. Integrating the content engineering practice and mindset into our ecosystems will begin to improve digital process and systems.
- As a content leader, take the reins and *drive the conversations* around intelligent customer experience and how content intelligence activates it. Start conversations with colleagues who are excited and passionate about the future of content.
- Share knowledge by *building or participating in communities* focused on the future of customer experience and intelligent content skill sets. Some examples include Society for Technical Communication (STC), conferences, online forums, and the [A] Content Order, which facilitate this kind of dialogue and idea-sharing. These kinds of groups can be built within an enterprise, as can committees, communities of practice, or centers of excellence.
- Begin the foundational work to *assess the current process* around content and *build working groups* to explore the future of how content is created and consumed.

Content engineers are working every day on new solutions for improved intelligent customer experiences. Through innovation and collaboration, we are all navigating the ever-expanding ecosystem that supports intelligence.

REGISTER NOW!

15–18 MAY 2020 AT THE HYATT REGENCY BELLEVUE

STC 2020 TECHNICAL COMMUNICATION SUMMIT
CONFERENCE & EXPO

Providing Value in an Agile Environment as a Technical Writer

Shutterstock/Jacob Lund

BY CINDY CURRIE | *STC Fellow* and
KIT BROWN-HOEKSTRA | *STC Fellow*

How do I provide value in an Agile environment?

With more and more companies using Agile methods for their software development, many technical writers are asking how they fit in to the team and program structures. Here's how it worked in one global IT company. The technical writer supported a cloud-based offering by writing the online help (for end-users) and the internal release notes (for delivery teams). The development teams worked two-week sprints in a 12-week program increment. Deployment to production occurred the last day of each sprint.

The technical writer worked alongside the architect and product owner to produce the online help, keeping pace with the sprint plans and documenting only what was produced in each sprint. Thus, she was providing an updated copy of the online help every two weeks. She updated the internal release notes, as well, but they were not deployed to production, as they were intended as a follow-up reference document to a training session on the offering delivery teams at the end of each program increment, right before release to market.

The technical writer was not considered to be on a "team," because she was located in China and working at the team level wasn't workable and really didn't make sense given what each team was focused on. It was the sum total of each team's output, once integrated, that created the vertical slice of value (in this case, a feature) that was her focus.

To do her work, the technical writer had access to the internal lab environment, the staging environment and a demo environment. As she developed expertise with the offering, she was called upon by the product manager to run customer demos, thereby showcasing her knowledge of the system, as well as the system itself. Customers were impressed by her ability to answer questions that the product manager had trouble answering. Several sales were progressed and/or closed. She increased her value to the entire program in this way.

In other cases, technical communicators are embedded into multiple scrum teams and produce content on a sprint+1 schedule. When localization is involved, the localized content is produced on a sprint+2 schedule. This structure works if you have structured content and a robust content management system, because these two things allow you to track chunks of content through the workflow and process them through localization in batches. Be sure to include your localization vendor in your planning and workflow, so that they know what to expect.

Our advice: Get in there and see where you fit in the overall team or program structure of your environment. Develop expertise with the product (as we technical communicators typically do), find a way to showcase it, and garner the admiration and respect of those around you due to the value that you add (what they perceive as value) to the program. Know your worth.

Ask a Tech Comm Manager is an advice column geared toward answering all those questions you have, but might be uncomfortable asking. We glean the questions from social media, forums, and most importantly, from you, dear reader. If we don't know an answer, we will interview experts and get information for you. Send us your questions to *kitbh.stc@gmail.com* or tweet them to *@kitcomgenesis* or the hashtag *#askTCmgr.*

Spaces, Doorways, and Usable Designs

BY KIRK ST.AMANT | *STC Fellow*

IT'S HAPPENED TO ALL OF US: You walk into a room and forget why you are there. It can be frustrating, and it's much more common than you might think. These situations reflect how our brains process information—something with implications for the usability of designs.

Recognizing Spaces

When we enter a space, we try to determine where we are, what we will do there, and what we can use to accomplish that objective. Achieving these goals involves the mental models we use to identify our location.

The process works like this: we have a mental picture—a "prototype of place"—that we associate with specific spaces. For example, when you hear "doctor's office," a picture comes to mind. When you hear "classroom," a different mental image appears. The idea is that we have different mental visual models we use to identify specific spaces. When we enter a new place, we attempt to match it to the mental models we have for different settings. Specifically, we scan the area to see how closely its features compare to those of the mental models we have for settings. The more closely the design of a space matches a given mental model, the more readily we can identify where we are.

Spaces and Actions

Once we know where we are, a separate mental process begins, one that tells us what we do in this space and what we can use to perform those activities.

These processes happen reflexively and are why we seem to know what to do in familiar spaces. They also explain why we feel disoriented when we can't identify where we are.

These prototypes of place are not innate; we learn them via repeated exposure over time. The more often I see a classroom as a place containing certain things (for example, desks and a chalkboard or whiteboard) organized in a particular way (for example, a chalkboard or whiteboard at the front of the room), the more my prototype of place for "classroom" associates those features with identifying that space. This is why we seem to act instinctively in spaces we know well but are often hesitant to act in new environments.

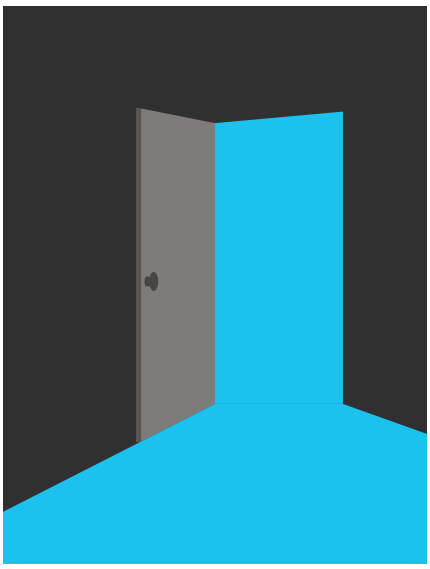
These same processes are key to effective interface design.

Interfaces and Usability

An interface is a virtual space. We identify "where" in a program we are based on the features on a screen. For example, if we are at the login screen of a program, we expect that interface to contain certain features—such as, a prompt for "username" and another for "password"—and be organized in a particular way—like the "username" prompt appearing above the "password" prompt.

These factors help us to identify where we are (login section of program), what to do there (enter username and password), and what to use (assigned username and password for that program). This is why we are able to log in to a variety of systems without needing to think about it. If we design virtual spaces to mirror what individuals expect them to look like, then they are easier to use.

But there is a catch to all of this.



Shutterstock/Andy Vinnikov

Doorways and Defaults

When we enter a space, we establish where we are and how to act. This requires us to recognize and remember where we are to keep our actions consistent and continuous. When we move to a new space, the brain needs to determine our new location and establish what to do there. Every time you enter a new space, however, your mind clears itself to establish your location and determine how to act there. This factor means moving from one space to another breaks the identification-action chain created by prototypes of place and requires us to "start over" again.

Doorways seem to cause this process. When you pass through a doorway—or an identifiable threshold marking the end of one space and the beginning of another—your mind initiates this "clear all" reflex, and you need to start the identification-action process over again. This is the "doorway effect" that causes many of us to forget what we were doing when we enter a new room. It also has implications for usable interface design.

This column examines how cognitive factors can affect technical communication and design processes. Email the editor at *stamantk@ecu.edu.*

Doorways and Designs

When we move from one screen in a computer program to another, we're moving through virtual spaces. Our journey through a computer program, therefore, could initiate the doorway effect and leave us wondering where we are and what we are doing when using software applications.

With these virtual spaces, it is not an actual doorway that initiates the effect: it is the design of the interface. If the configuration of the screen shifts or is associated with some visual change (for example, it goes black when transitioning from one screen to the next), the doorway effect could be initiated. This factor could affect how easily individuals use items based on how quickly they can orient themselves in a new digital space.

How can we avoid this factor? First, keep the design of the interface as consistent as possible when moving from screen to screen in a system. The more a new screen looks like a preceding one, the less likely individuals are to view it as a new space or to note a transition (in virtual spaces) has occurred. Next, mitigate or eliminate visual cues that could indicate moving through different digital spaces. This could be as simple as minimizing the time it takes for a new page to load to avoiding "transitional" effects (for example, an hourglass icon against a blank screen to indicate "wait for new screen").

That's not to say always avoid such designs, but rather to use them conscientiously. They can be essential to making users aware of key transitions and to making essential re-orientations. Such transitions mid-process, however, could cause problems. Specifically, they could prompt users to forget what they are doing mid-process, and that factor could affect the success and satisfaction of the user's experiences.

Final Thoughts

The mind is a complex data processor. The better we understand its workings, the more effectively we can design to match its operations. By understanding how our minds process space, we can create designs that enhance the user experience—whether it is in physical or virtual spaces.

RESOURCES

Brenner, Charles B., and Jeffrey M. Zacks. "Why Walking through a Doorway Makes You Forget." *Scientific American*. 13 December 2011. *https://www.scientificamerican.com/article/why-walking-through-doorway-makes-you-forget/.*

Eyal, Nir. *Hooked: How to Build Habit-Forming Products*. Penguin: New York, NY, 2014.

St.Amant, Kirk. "Reflexes, Reactions, and Usability: Examining How Prototypes of Place Can Enhance UXD Practices." *Communication Design Quarterly*, 6.1 (2018): 45–53.

Transform your career.

Renew with STC today.

HIGH-QUALITY EDUCATION

TRUSTED RESOURCES

EXPERIENCED CONNECTIONS

PROFESSIONAL CERTIFICATION

stc.org

+1 (703) 522-4114

membership@stc.org

STC

The Components of Content: Meet Patrick Bosek

BY SCOTT ABEL | *STC Associate Fellow*

EACH YEAR MY FIRM, The Content Wrangler, surveys technical communication teams around the globe to learn how they create, manage, translate, localize, and deliver technical content. Over the past decade, we've spotted a number of trends—XML authoring, single-source multi-channel publishing, and the move to dynamic publishing—that have positioned the discipline of technical communication at the forefront of publishing innovation.

Modern technical documentation shops use a variety of approaches, standards, and tools to accomplish their work. Not every techcomm team or squad works in the same manner. Differences in strategy, methodology, and software platforms vary for a variety of reasons. Some teams lag behind. Others are way out in front, crafting innovative content solutions that other parts of the organizations that they serve recognize, respect, and sometimes want to replicate.

In this month's installment of "Meet the Change Agents," I interview technical content management maven Patrick Bosek, Co-Founder and Chief Executive Officer of Jorsek, Inc. makers of easyDITA, a component content management system used by technical communication teams.

Scott Abel: Patrick, thanks for taking the time to speak with me about the issues impacting the effective and efficient management of technical documentation content. For those readers who don't know who you are or what you do for a living, can you take a moment to tell us about yourself and your connection to the field of technical communication?

Patrick Bosek: Thanks, Scott. I think your intro covers who I am and where I work. In terms of my connection to tech comm, it's where I've worked for most of my career. I've been working to build tools for technical writers for about 10 years now. It's a form of content that feels really worthwhile to me. I started my career dealing with more marketing-focused content, websites and such, but I found that dealing with knowledge-oriented content was a lot more interesting to me, so here I am.

SA: What do you think are the biggest challenges facing technical documentation teams today?

PB: Becoming something new. We've known for a while that tech comm was going to look different in the future, and I think that future is here (even if all of those differences are not evenly distributed across all organizations). Technical writers need to become information developers. The skills are largely the same, but the focus is more on building knowledge and information experiences.

The biggest practical change might be that information developers

In the digital age, change happens quickly. This column features interviews with the movers and shakers—the folks behind new ideas, standards, methods, products, and amazing technologies that are changing the way we live and interact in our modern world. Got questions, suggestions, or feedback? Email them to *scottabel@mac.com*.

can be measured on their efficacy, whereas technical writers are generally only measured on their throughput (if they're measured at all). Because information developers generally publish content directly to the places users consume it, they can measure use and engagement. This is a new, and significant, change for an industry that has been built around non-measurable deliverables like books and hard copy PDFs. And this trend is only going to accelerate as content and knowledge systems become better connected. In the near future, you'll be able to cross-check a user's content interactions with their support requests and system use. When you have that level of data available, you'll be able to truly measure the impact of information developers and that knowledge can help you drive improvements.

SA: How has content management changed over the past decade? Have there been any major innovations or technological advancements that have (or perhaps, will) change the way content management systems are designed or what they're capable of helping us do?

PB: Wow, decade? Yeah, a ton has changed in the past decade. I think the biggest change is the shift to the cloud. Ten years ago, cloud was a small portion of the content management system (CMS) landscape, and today I think it's the default. And it makes a lot more sense that way. From a pure cost perspective, internal IT management of CMS technology is expensive and generally unnecessary, especially when you're looking at specialized CMS technology like component content management systems (CCMSs). And from a strategic perspective, the cloud enables connection. Legacy on-premise systems are more difficult to connect, and isolated content systems can't offer as much value as those in the cloud.

On the topic of component content management, it's been around for more than 10 years, but the real traction has been more recent. Today, smart companies realize that having a CMS is not sufficient, the total content ecosystem requires multiple major components, many of which broadly fall under the umbrella of "CMS." Good CCMSs act as the true center point for knowledge in an organization. Since CCMSs are generally built around presentation-agnostic, structured content (most often XML), they can be the system of record for knowledge and the source for all end user systems. End user systems can be Web systems like documentation portals, knowledge bases, chatbots, websites, or learning management systems, but they can also be PDFs or printed copies. The reality is that almost all organizations have knowledge content deployed to many places, both internally and externally, and this will never change, because different use cases demand different delivery systems, but the core content across these systems needs to be consistent. The only way to achieve this is with a good CCMS.

SA: In our annual survey of technical communication professionals, we ask respondents what their biggest challenges are. One of the biggest challenges facing technical communication teams are inadequate or inappropriate tool sets. Words used to negatively describe technical communication tools are common, and those tools are often are characterized as "outdated," "difficult," "cumbersome," and "old school." Many of the most vocal respondents complain that vendors could provide better experiences, but it doesn't seem to be a priority.

Your professional biography says you are a "skilled developer, thoughtful manager, and passionate customer advocate." What lessons have you learned about serving your customers and what changes have you made to your product, easyDITA, to address their concerns about usability, maintainability, and scalability?

PB: We're constantly iterating on easyDITA with the intention of providing a better user experience. It's a challenging process. Technical content is itself often quite complex, thus building tools that can handle this complexity while still providing an intuitive experience to a broad range of users is challenging. In the process of doing this, I think the biggest lesson we've learned is how to balance adaptability and focus. easyDITA is a platform for everyone in the information development workflow. Technical writers need a different experience than SME-reviewers, for example. Our primary focus now is providing the right experience for the right user. A technical content platform doesn't get to be a single experience system.

SA: You've written and presented about several of my favorite content topics over the years. I always enjoy your take on a subject. I find your views on what makes a "good content creation system" (the subject of a blog post you authored on the easyDITA blog) of interest, and I'm certain our readers will as well.

Can you share with us your thoughts on what really matters when creating a content production platform? What are the pillars of good content creation?

PB: I think the article you're referring to (*https://easydita.com/3-pillars-of-a-good-content-creation-system/*) does a good job of outlining the three pillars, which are efficiency, flexibility, and scalability.

I think the term scalability as it relates to content systems needs a bit of reframing. People tend to think of scalability as how much stuff can be stored in or moved through a technology, but to me this definition is out of date. Quantity scalability is certainly still important, but when you're considering modern, tier one applications, it should be table stakes, or an implementation detail, rather than a key deciding factor between one tool and another.

The scalability that matters is the ability to scale the whole system: people, process, and technology. When you start looking at scalability this way, you realize that the methods of collaboration and organization matter a lot more than how many gigabytes your CMS can store.

On a related note, Mark Baker left a very interesting comment on a blog post about a year ago (*https://medium.com/@mbakeranalecta/david-there-are-two-fundamental-modes-of-collaboration-showing-and-hiding-3a54397e0bcc*). While I don't agree with his point 100 percent, the basic premise that approaching collaboration as an open system doesn't scale is something I strongly agree with. Collaboration needs structure to scale. Really everything needs structure to scale, it's a basic law of nature; it's true in skyscrapers, and it's true in content and knowledge delivery.

SA: There's been a movement among a subset of technical communicators to promote and adopt the lightweight markup language known as Markdown. As someone whose career has been based in part on helping large organizations move from creating unstructured content to crafting modules of structured, semantically rich, intelligent content using XML markup, the idea of working in Markdown seems both nonsensical and ill-conceived.

I realize creating semantically rich, intelligent, XML content is not the solution to every problem, but intelligent content solves many problems, and when implemented correctly, it helps organizations do things with their content that seemed previously unimaginable.

I worry that those who aim for working in Markdown are moving in the wrong direction, especially as we're moving headfirst into the Fourth Industrial Revolution where content must be machine readable and processable, in addition to being fine-tuned for the humans we hope will consume it.

Can you help me understand the pros and cons of Markdown? What things do technical writers need to consider before they adopt Markdown?

PB: There is so much to say on this topic. You could write an entire book about it (and some have). Before I directly answer your question, I think it's best to reframe it a little bit. I think it's more accurate to say people are promoting and adopting docs-as-code, rather than just Markdown. Markdown is really just the doc type that fits into docs-as-code best, so on its own, there is little reason to consider it for technical documentation. It's basically just wiki text. Not that there is anything inherently wrong with that, it's just to say that its place in a docs-as-code ecosystem is its strength more than any innate property.

Looking at docs-as-code using Markdown, there are definitely pros and cons. I think the docs-as-code methodology has a lot of significant benefits, but I think representing it as the only future for all of technical communication (as some people are right now) is not only invalid, it's also dangerous.

I see this debate happening in many channels: In some places it's Markdown vs. DITA, in some places it's docs-as-code vs CCMSs or "traditional tools," but it's basically the same debate.

My perspective is that there is no "single correct format and method." The reality is that docs-as-code attracts people in a bubble of technical writers who are not only in software-focused organizations but are often in software organizations with a start-up mentality, which creates a set of circumstances that naturally lend themselves heavily to docs-as-code.

API and single-product, low-complexity reference content is best handled with a docs-as-code methodology, but that's a pretty small subset of tech writing, and if you try to extrapolate to all cases—even all software-focused cases—you're going to run into problems.

The more content you have, the more semantics you need to drive high-quality search. The more varied your audience and product configurations, the more you need strong reuse and filtering mechanisms—if you care about reader experience, that is. The more places you need to deliver your content, the more consistent it needs to be, and the structure provided by DITA/XML is extremely helpful for this consistency.

Markdown and docs-as-code is great if your situation is characterized by:

- A single product
- A single audience
- A single language
- A single (or very few) target outputs
- Simple metadata requirements

DITA and CCMSs are often a better choice if your situation is characterized by:

- Multiple products sharing content
- Multiple audiences
- Many languages
- Complex output requirements
- Complex or significant metadata requirements

These aren't mutually exclusive. Your organization may have both scenarios at different times, different points in the development and maintenance lifecycle, or different product groups.

One is not better than the other; they are just different. I think we're doing everyone a favor by helping each other understand the right match between tools and situations, rather than approaching this subject as dogma.

SA: A thoughtfully chosen product name can differentiate a product from its competitors and quickly help consumers choose the product that they believe is right for them. While I don't understand why a CMS vendor would include the name of a standard (for example, DITA) in their product

REGISTER FOR LIVE WEBINARS (2 CEUs)

Behind-the-Scenes Networking: The Secret to Today's Job Search

December 11, 2019
3:00 – 4:00 PM EDT

INSTRUCTOR:
Alyssa Fox

COST:
Members: $59
Nonmembers: $99
Student Members: $29

2018-2019 STC Salary Database: Utilizing the Information and Key Trends

December 18, 2019
1:00 – 2:00 PM EDT

INSTRUCTOR:
Michael Becher

COST:
Members & Student Members: FREE
Nonmembers: $99

www.stc.org

name (what happens if another standard takes hold in the future?), I love that you included the word "easy" in your product name.

As you attempt to convince those outside of the technical communication sector of the importance of creating, managing, and delivering semantically-rich intelligent content, do you think your product name gets in the way?

PB: *(Laughs)* Our product name has been the subject of much debate. We chose it because we wanted something straightforward, simple, and easy to remember. I think it is that. At the same time, we've had people choose other systems, because the name didn't sound "heavy duty enough"—that's a quote.

SA: Chatbots and conversational artificial intelligence provide big opportunities for technical communication professionals. You've written that you believe that chatbots are hard to get right, even harder than autonomous vehicles. Can you explain your thinking on this subject?

PB: Sure. Let's start by qualifying that comparison a bit more. When I say I think chatbots might be harder than autonomous vehicles, what I'm really saying is that I think creating—and I use that term very consciously—a human-equivalent computer system to perform a task is easier for autonomous vehicles than chatbots. It's meant to be a bit cheeky and provocative to drive a point.

Notice that I carefully use the term "creating" and not "deploying" here. Autonomous vehicles are much higher consequence than chatbots, so deploying them is infinitely harder than building them, as deploying autonomous vehicles requires vastly more effort in testing and regulating.

I believe the best quote I've heard about autonomous vehicles came from an engineer working on a vehicle for DARPA Grand Challenge. It's going back a while, now, so I can't find the link, but he basically said, "People want to call this AI, but that's not really an accurate characterization; what we're building is just really advanced calculators."

This quote shaped a lot of my thinking about AI over the years. Everyone wants their thing to be AI, because they perceive that it's worth more money when it is, but many of the things we're calling AI aren't really AI.

This matters, because I don't think you need true AI to create a human-equivalent autonomous vehicle (they've existed for some time now), but I think you might need true AI to really produce a human-equivalent chatbot.

There are fairly well-established rules for driving a car, you can conceive of how these rules can be implemented into a system, and I don't think you really need true AI to do it. But when you think about deploying human-equivalent chat technology, you have to remember that the rules are much fuzzier. Human interaction is incredibly complex. This complexity might require true AI, which we don't have yet.

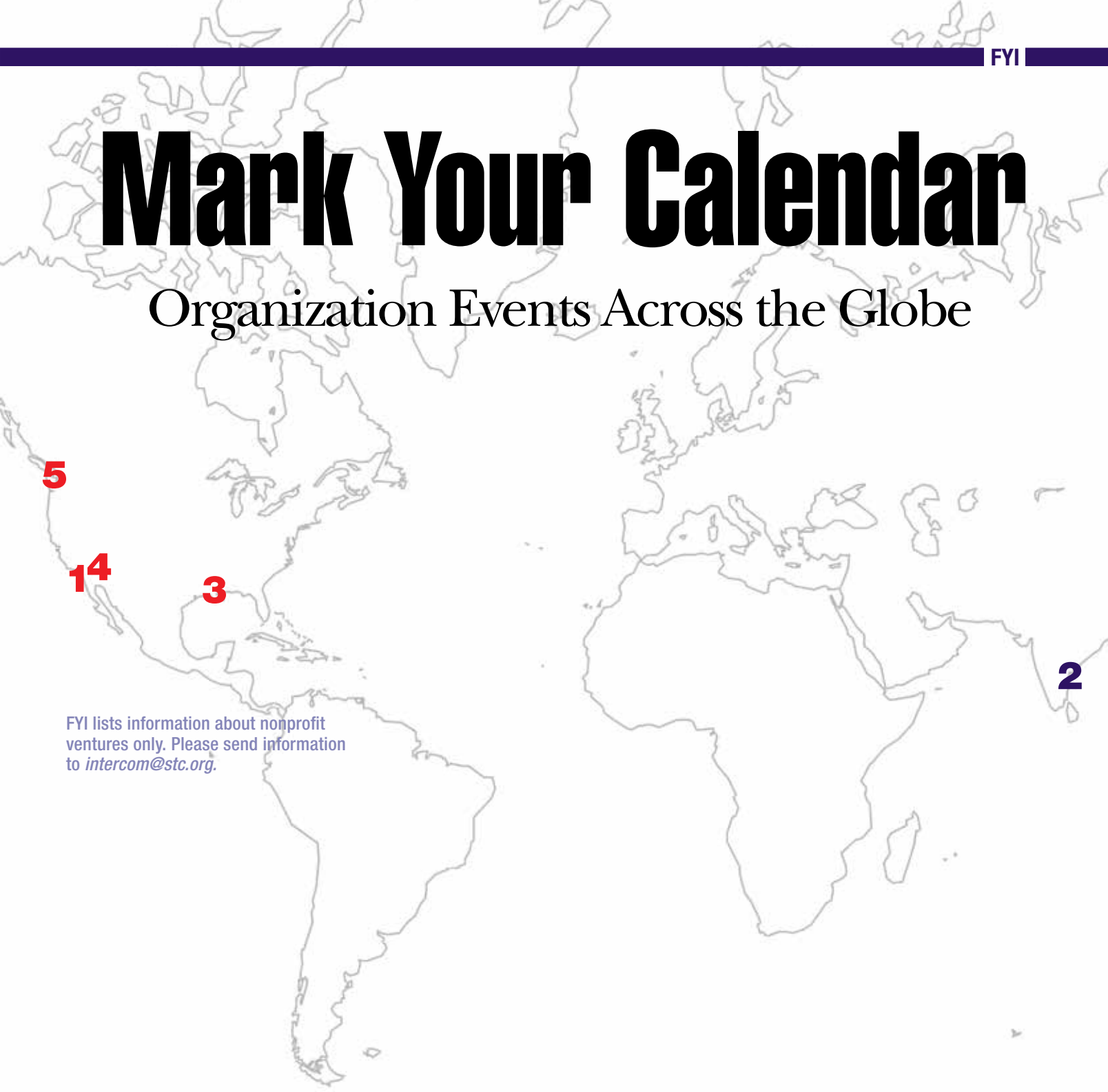
All of that said, I don't think you necessarily need a human-equivalent chatbot to have a useful, chat-style content system. My real point with this comparison is to warn people that chatbot systems aren't plug-and-play, they're not magic, and they require a lot of thought and work to get right.

SA: I'm afraid we're out of time. Thank you on behalf of our readers for taking a few moments to share a little about yourself and your views on technical communication content.

PB: Any time.

Mark Your Calendar

Organization Events Across the Globe



FYI lists information about nonprofit ventures only. Please send information to *intercom@stc.org.*

1 6–9 Nov 2019
The **American Medical Writers Association** (AMWA) Medical Writing and Communication Conference will take place 6–9 November 2019 at the Sheraton San Diego Hotel and Marina in San Diego, CA.
AMWA
conference@amwa.org
240-238-0940
https://www.amwa.org/page/Conference

2 5–7 Dec 2019
The **India Chapter of STC** will hold its 21st Annual Conference in Chennai, India. For more information, contact STC India.
https://stc-india.org/conferences/2019/

3 2-5 Jan 2020
The 94th Annual meeting of the **Linguistic Society of America** will take place 2-5 January 2020 at the Hilton New Orleans Riverside in New Orleans, LA.
LSA
https://www.linguisticsociety.org/event/lsa-2020-annual-meeting

4 27-30 Jan 2020
The annual **Reliability and Maintainability Symposium** (RAMS) will be held at the Marriott Renaissance in Palm Springs, CA.
RAMS
http://rams.org/
rams2020@rams.org

5 13-16 Feb 2020
The **American Association for the Advancement of Science** (AAAS) annual meeting will be held at the Washington State Convention Center in Seattle, WA.
https://meetings.aaas.org/
meetings@aaas.org

* STC-related event

Wearing Many Hats

BY SURAG RAMACHANDRAN | *STC Member*

I WORK AS AVP and Head of Product Content and Learning at MBB Labs, the global innovation center for Maybank, the largest bank in Malaysia. I work in Bangalore, India, in a research and development wing that functions like a startup. I was among the first fifty employees, and the company was looking for people with multiple skill sets to join them. In this role, I oversee technical documentation, marketing communication, (marcomm), and learning.

Before joining Maybank, I worked at Oracle. At Oracle and other multinational corporations, the three roles I oversee are performed by three different professionals. However, since the Bangalore wing of Maybank was just beginning, Maybank was looking for people who could take on multiple roles. I was used to managing technical documentation at Oracle as Senior Documentation Manager, but marcomm and learning management were new to me.

Becoming familiar with marcomm and learning management from a technical writing background made me realize that other technical communicators could also adapt to take on additional responsibilities. When I was asked to form a team, I looked for technical writers who were interested in expanding into new areas. I explained the job role to candidates, and hired ones who were really enthused about the prospect of learning new things. Each hire is unique, and I am grateful that the three team members I hired are capable and motivated.

In marcomm, one of our major deliverables is videos, especially explainer videos. We started doing videos because we found that the decision makers may not have two hours to read a document, but can find time for a two minute video. Explainer videos are short and focused. For the technical writers I hired, creating full-fledged documents and distilling content for short, crisp videos was already part of their skill set. Writing the script and narration and getting them approved was also the part of the process.

In learning, it's important to get the outline and structure of the deliverables ready before proceeding with the content, quite similar to the process of writing a user guide. While user guides "guide" the user through a process, structured learning content "trains" the user to perform a task.

The disadvantage of performing multiple roles is that you really have to be super productive and not let one responsibility impact the others. However, the advantages to having multiple skill sets far outweigh the disadvantages. Apart from being able to learn many new things and interact with multiple people in different roles, you also have many different career paths and options to pursue.

These same things motivated my team members to join my team and perform their responsibilities. To say that I am proud of my team is an understatement. We have achieved a lot in very little time, and we're really looking forward to doing more in the future.

Shutterstock/Stmool

CERTIFIED PROFESSIONAL

Technical Communicator

CPTC™

FOUNDATION

- **Project Planning**
- **Project Analysis**
- **Content Development**
- **Organizational Design**
- **Written Communication**
- **Reviewing & Editing**
- **Visual Communication**
- **Content Management**
- **Production & Delivery**

CONTACT

For more information about certification and to start the process, visit *www.stc.org* or email *stc@stc.org.*

Certified Professional Technical Communicator™ (CPTC)

Advance your Career and the Profession Earn the CPTC Foundation Credential Today

The Certified Professional Technical Communicator (CPTC) credential assures employers, colleagues, and the public that you have the knowledge and skill to handle complex technical communication projects from the project planning stage through production and delivery.

Benefits

Why earn the CPTC credential? Because the CPTC credential:

- Distinguishes you from your peers;
- Shows you have the most up-to-date knowledge in the field;
- Opens up job opportunities and enhances job mobility;
- Elevates the profession;
- Gives you a sense of pride in your career; and
- Demonstrates your commitment to the field.

The CPTC credential gives you the edge you need to stand out to employers.

Continuing Education Requirements

Points may be obtained the following ways:

Event	Points
STC Annual Membership (any membership type for Foundation certificants)	2
STC Recorded Webinar (self-study)	1
STC Live Educational Webinar (free, sponsored, and community webinars excluded)	2
STC Online Courses	6
STC Summit Pre-Conference Courses (full day)	6
STC Summit Pre-Conference Courses (half day)	3
STC Annual Summit	8
Begin and complete a college-accredited course related to the Technical Communication field	8
Published articles that relate to any aspect of *Technical Communication* (2/article)	2
Published books publicly available on topics related to *Technical Communication* (5/book)	5
Presentations at conferences related to aspects of *Technical Communication* (2/presentation)	2
Total needed within 2 years post-certification date	**12**

Fees

Exam fees: STC Members $250, Non-Members, $495

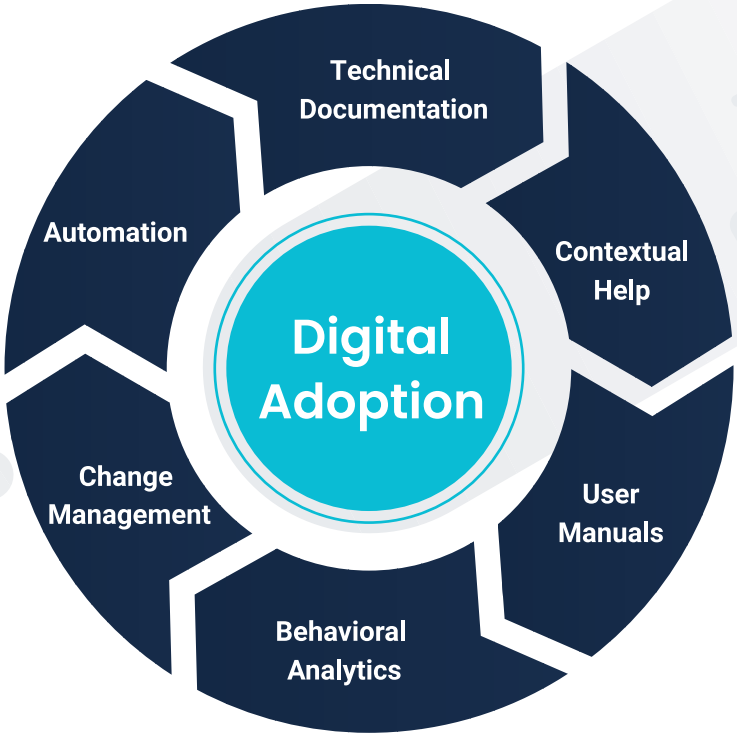
Be a leader. ***Take your career to the next level by obtaining your credential. It's the most efficient way to prove your skills and knowledge in the technical communication field.***

Advance your Career and the Profession — Earn the CPTC Foundation Credential Today!
Learn more at www.stc.org/certification

whatfix

Are You Still Using Traditional Product Help?

Whatfix helps you transform your product documentation, making it interactive, engaging and available in real-time within the app.



How Does Whatfix Help You?

- Compatible with all web-based applications
- Efficient multilingual authoring, instant publishing and editing
- Measure your product documentation effectiveness
- SCORM compliant content production available instantly in multiple formats like Pdfs, videos, slides, and smart URLs

AVNET · CardinalHealth · SOPHOS · ManpowerGroup · cisco · bmc

sales@whatfix.com | +1800-459-7098 | @whatfix